

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
3η Εργασία - Τμήμα: Αρτίων Αριθμών Μητρώου
Κ22: Λειτουργικά Συστήματα – Χειμερινό Εξάμηνο '19
Ημερομηνία Ανακοίνωσης: Δευτέρα 25/11
Ημερομηνία Υποβολής: Τρίτη, 17/12 και Ώρα 23:59

Εισαγωγή στην Εργασία:

Ο στόχος της εργασίας είναι να δημιουργήσετε ανεξάρτητα προγράμματα τα οποία *τρέχουν ταυτόχρονα* και προσομοιώνουν την λειτουργία ενός Σταθμού Λεωφορείων. Καλείστε να υλοποιήσετε τουλάχιστον 3 τύπους διεργασιών που η κάθε μία έχει διαφορετικό ρόλο. Οι τρεις τύποι διεργασιών είναι:

1. Λεωφορεία (buses) που επιθυμούν να μπουν στο σταθμό, να αποβιβάσουν επιβάτες, να μείνουν για κάποιο πεπερασμένο χρονικό διάστημα και τέλος αφού παραλάβουν επιβάτες να αποχωρήσουν. Τα λεωφορεία ταξιδεύουν σε διαφορετικές περιοχές της χώρας που είναι: i) Αττική, Στερεά & Κεντρική Ελλάδα (ASK), ii) Πελοπόννησο (PEL), και iii) Βόρειο Ελλάδα (VOR).
2. Διευθυντής σταθμού (station-manager).
3. Επιτηρητής λειτουργίας σταθμού (comptroller).

Οι παραπάνω τύποι διεργασιών είναι ανεξάρτητα προγράμματα τα οποία θα μπορούν να *τρέχουν ταυτόχρονα* και να *υλοποιούν συνεργατικά* το τι συμβαίνει στο χώρο ενός σταθμού λεωφορείων.

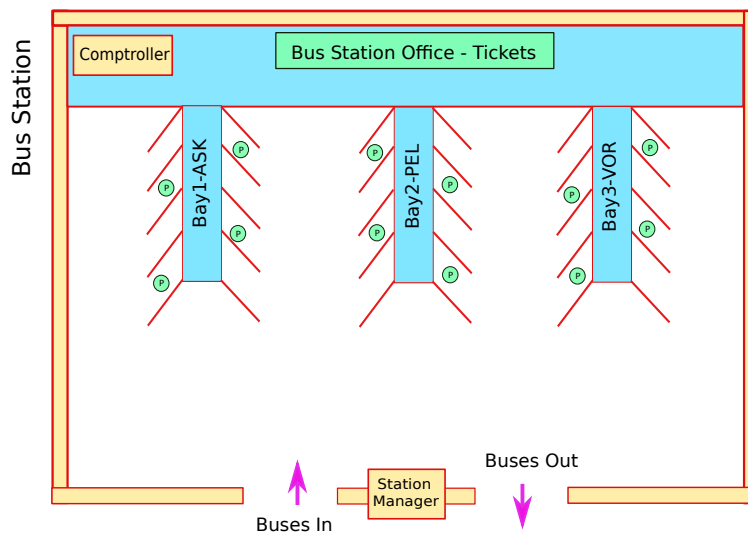
Κάθε λεωφορείο που φτάνει σε τυχαία χρονική στιγμή θέλει να εισέλθει στο σταθμό, να σταθμεύσει στην νησίδα που θα του υποδειχθεί, να αποβιβάσει επιβάτες, να περιμένει για ένα χρονικό διάστημα, να επιβιβάσει το κοινό που έχει αγοράσει εισιτήρια, και να αποχωρήσει. Πριν εισέλθει στο σταθμό, κάθε λεωφορείο θα πρέπει να συνεννοηθεί με τον station-manager ώστε να πάρει το σχετικό οκ αλλά και να πληροφορηθεί για την νησίδα που τελικά θα μπoreσει να σταθμεύσει.

Το Σχήμα 1 παρέχει ένα διάγραμμα της γενικής λειτουργίας του Σταθμού. Όταν ένα λεωφορείο εισέρχεται, εξέρχεται, ή κάνει μανούβρα στο χώρο του σταθμού θα πρέπει να είναι πάντα το πολύ μαζί με ακόμα ένα λεωφορείο που βρίσκεται σε κίνηση μέσα στο περιορισμένο αυτό χώρο. Αφότου ο station-manager σιγουρέψει ότι δεν υπάρχει υπερβολική κίνηση στο χώρο του σταθμού, δίνει το οκ σε ένα από τα πολλά λεωφορεία που περιμένουν εκτός σταθμού να προχωρήσει. Το εν λόγω όχημα εισέρχεται, σταθμεύει, και μένει στην συγκεκριμένη θέση που του έχει δοθεί για όσο χρονικό διάστημα χρειάζεται για να ολοκληρώσει την διαδικασία εκφόρτωσης/φόρτωσης επιβατών.

Ο ρόλος του station-manager είναι να επιβλέπει την ασφαλή λειτουργία της υποδομής και να καταγράφει σε ένα ημερολόγιο (reference ledger) όλη την δραστηριότητά που λαμβάνει χώρα στο σταθμό. Ανάλογα με την ώρα άφιξης και την περιοχή ταξιδιών του λεωφορείου, για κάθε όχημα, στο reference ledger αναγράφονται τα σχετικά στοιχεία όπως ώρα άφιξης, αριθμός πινακίδας, τύπος λεωφορείου, τελική θέση (ή νησίδα) στάθμευσης, αριθμός επιβατών που αποβιβάζονται και status του οχήματος που έχει αφιχθεί (δηλ. όχημα παρόν στο σταθμό).

Όταν ένα λεωφορείο πρέπει να αναχωρήσει και αφού έχει επιβιβάσει έναν αριθμό απο επιβάτες, ακολουθεί την αναμενόμενη ακολουθία γεγονότων: έρχεται σε συνεννόηση και περιμένει να πάρει το οκ από τον station-manager. Όταν ο τελευταίος εκδώσει εντολή αναχώρησής (αφού ξεκαθαρίσει ότι δεν υπάρχει άλλη εξερχόμενη/εισερχόμενη κίνηση), στο ledger αναγράφονται η ώρα της εν λόγω αναχώρησης του οχήματος, αριθμός επιβατών, και αλλαγμένο status (δηλ. όχημα έχει αποχωρήσει από σταθμό).

Ο comptroller είναι ένα ανεξάρτητο πρόγραμμα που σε τακτά χρονικά διαστήματα παρέχει την κατάσταση στο Σταθμό (δηλ. ποια και πόσα λεωφορεία είναι παρόντα). Επίσης με μια περιοδικότητα που ορίζεται από τον χρήστη



Σχήμα 1: Διάγραμμα για την λειτουργία του Σταθμού Λεωφορείων.

παρέχει στατιστικά ενδιαφέροντος όπως π.χ. σύνολο επιβατών, αριθμό αποβιβασθέντων, αριθμό επιβιβασθέντων, μέσο όρο παραμονής οχημάτων στο σταθμό, αριθμό διαθέσιμων χώρων στάθμευσης σε συγκεκριμένη χρονική στιγμή, και το μέσο όρο αναμονής των οχημάτων στο σταθμό.

Για δική σας διευκόλυνση, μπορείτε να δημιουργήσετε ένα πρόγραμμα (`mystation`) που αρχικοποιεί τα δεδομένα που απαιτούνται, δημιουργεί τις απαραίτητες διεργασίες με την βοήθεια κλήσεων `fork()` και `exec*()`. Το πρόγραμμα `mystation` μπορεί επίσης να δημιουργεί τους semaphores, το shared segment, και οτιδήποτε άλλο χρειάζεται. Στο shared segment θα πρέπει να είναι αποθηκευμένα τα στοιχεία του reference ledger (η τμήμα του). Εναλλακτικά για τις προτεινόμενες διεργασίες `mystation`, `station-manager`, `bus`, `comptroller` μπορείτε να χρησιμοποιείτε πολλαπλά `ttys` για να ξεκινήσετε χειρωνακτικά οποιαδήποτε διεργασία (ή στιγμιότυπο) επιθυμείτε να έχετε στην λύση σας.

Στην άσκηση θα πρέπει να:

1. χρησιμοποιήσετε ένα σετ από σηματοφόρους ώστε να έχετε μια επιτυχή συνεργασία μεταξύ των ανεξάρτητων διεργασιών, shared memory segment για την σωστή εξυπηρέτηση των οχημάτων, όπως επίσης και να δημιουργήσετε σχετικές δομές όπου αυτές χρειάζονται,
2. έχετε όλες τις διεργασίες για να προσαρτήσουν το παραπάνω shared memory segment ώστε να μπορούν να προσπελάσουν περιεχόμενα ενδιαφέροντός τους,
3. χρησιμοποιήστε POSIX Semaphores για να υλοποιήσετε τη λύση σας που θα εμπεριέχει σχετικές κλήσεις `P()` και `V()`,
4. παρέχετε τη δυνατότητα τα προγράμματά σας να παραμείνουν 'απασχολημένα' δηλ. να κάνουν `sleep()` για περιόδους χρόνου που καθορίζονται με παραμετρικό τρόπο.

Θα πρέπει να επιδείξετε την ορθότητά της λύσης σας όσον αφορά στην ταυτόχρονη εκτέλεση των διεργασιών που λαμβάνουν μέρος στην λύση που δίνετε.

Διαδικαστικά:

Το πρόγραμμά σας θα πρέπει να γραφτεί σε `C` (ή `C++` αν θέλετε αλλά χωρίς την χρήση STL/Templates) και να τρέχει στις μηχανές Linux workstations του τμήματος.

- Υπεύθυνοι για την άσκηση αυτή (ερωτήσεις, αξιολόγηση, βαθμολόγηση κλπ.) είναι η κ. Ελένη Μαντζάνα

cs2180007+AT-di, ο κ. Valentin Ivanov cs2180001+AT-di, και ο κ. Μάνος Τορόακης cs2180016+AT-di.

- Παρακολουθείτε την ιστοσελίδα του μαθήματος <http://www.di.uoa.gr/~ad/k22/> για επιπρόσθετες ανακοινώσεις αλλά και την ηλεκτρονική-λίστα (η-λίστα) του μαθήματος στο URL <https://piazza.com/uoa.gr/fall2019/k22/home>
- Το πρόγραμμα σας (source code) πρέπει να αποτελείται από **τουλάχιστον τρία** (και κατά προτίμηση πιο πολλά) διαφορετικά αρχεία. Το πρόγραμμά σας θα πρέπει **απαραιτήτως να κάνει χρήση separate compilation**.

Διατύπωση του Προβλήματος:

Το Σχήμα 1 παρέχει μια προτεινόμενη οργάνωση διεργασιών που δουλεύουν *ταυτόχρονα* για να επιτύχουν την λειτουργία του Σταθμού Λεωφορείων. Μπορούμε να υποθέσουμε ότι ο σταθμός εξυπηρετεί οχήματα που ταξιδεύουν στις τρεις γεωγραφικές περιοχές που έχουμε αναφέρει: ASK, PEL, VOR.

Οποιοδήποτε λεωφορείο παρκάρει κάτω από την υπόδειξη και διαχείριση του station-manager. Αν υπάρχουν πολλαπλά οχήματα που θέλουν να εισέλθουν στο σταθμό, ο station-manager τα εξυπηρετεί με την παρακάτω πολιτική: τα οχήματα VOR έχουν προτεραιότητα, ακολουθούν εκείνα που είναι από την ASK και τέλος εκείνα της PEL. Αν οχήματα από τις περιοχές VOR/ASK δεν έχουν χώρο για στάθμευση στην δικιά τους νησίδα μπορούν να σταθμεύσουν στην νησίδα των PEL εάν φυσικά υπάρχουν εκεί διαθέσιμες θέσεις. Οχήματα PEL μπορούν να σταθμεύσουν μόνο στην νησίδα τους. Γενικά, ο station-manager καθορίζει την θέση που τελικά παίρνει κάθε λεωφορείο σύμφωνα με τους παραπάνω κανόνες.

Όταν ένα όχημα λάβει το ok από την station-manager, παίρνει την θέση του και αρχίζει να αποβιβάζει επιβάτες ενώ αναμένει μέχρι να επιβιβαστούν όλοι όσοι έχουν εισιτήριο για τον επόμενο ταξίδι. Όταν η παραπάνω διαδικασία ολοκληρωθεί, το λεωφορείο ενημερώνει τον station-manager ότι θα ήθελε να αναχωρήσει. Ο τελευταίος δίνει την εν λόγω εντολή αποχώρησης εφόσον δεν υπάρχει υπερβολική κίνηση εντός σταθμού και όλες οι σχετικές για το λεωφορείο πληροφορίες συμπεριλαμβανομένου και του νέου αριθμού επιβατικού κοινού καταγράφονται στο ledger.

Συμπεριφορά Διεργασιών

Η διεργασία *bus* θα πρέπει να εκτελεί τις παρακάτω λειτουργίες:

1. Άφιξη και αναμονή έξω από σταθμό για εξυπηρέτηση από το station-manager.
2. Ενημέρωση από station-manager για ασφαλή είσοδο στο σταθμό και σε συγκεκριμένη νησίδα.
3. Άφιξη στην νησίδα που έχει οριστεί και αποβίβαση.
4. Ενημέρωση ledger για τον αριθμό των αφίξεων.
5. Παραμονή στο χώρο στάθμευσης και επιβίβαση επιβατών.
6. Αίτηση για αναχώρηση από σταθμό και αναμονή για την εξυπηρέτηση από το station-manager.
7. Ενημέρωση από station-manager για ασφαλή αναχώρηση.
8. Αναχώρηση και είσοδο στο δημόσιο δρόμο.

Η διεργασία *station-manager* εκτελεί τις παρακάτω λειτουργίες:

1. Δέχεται αιτήσεις για είσοδο στο σταθμό από πιθανώς πολλαπλά λεωφορεία. Αν στο σταθμό υπάρχει το πολύ ένα εξερχόμενο όχημα, δίνει εντολή σε ένα από τα οχήματα που περιμένουν εκτός σταθμού, να παρκάρει σε συγκεκριμένη διαθέσιμη θέση χρησιμοποιώντας την πολιτική τοποθέτησης.
2. Δέχεται αιτήσεις για αναχώρηση πιθανά από πολλαπλά λεωφορεία και αν δεν υπάρχει ένα το πολύ εισερχόμενο όχημα σε κίνηση μέσα στο σταθμό, δίνει εντολή για αναχώρηση σε ένα από τα λεωφορεία που αναμένουν την φορά.
3. Και στις δυο παραπάνω περιπτώσεις ενημερώνει το public ledger για την ώρα εισόδου/αποχώρησης από

σταθμό συγκεκριμένου οχήματος και την κατάσταση του (παρουσία/αναχώρηση), χρονικά δεδομένα, και θέση στάθμευσης διαθέσιμη/απασχολημένη,

4. Σε οποιαδήποτε χρονική στιγμή υπάρχουν το πολύ ένα εισερχόμενο και ένα εξερχόμενο όχημα που κινούνται την ίδια χρονική περίοδο στο χώρο του σταθμού.

Η διεργασία comptroller:

1. Σε τακτά χρονικά διαστήματα παρέχει την κατάσταση που επικρατεί στο σταθμό: παρέχει τον ακριβή αριθμό οχημάτων που έχουν σταθμεύσει, τις θέσεις που είναι διαθέσιμες και τον αριθμό επιβατών που τα εν λόγω οχήματα έχουν αποβιβάσει.
2. Επίσης δίνει στατιστικά για την μέση αναμονή, αναμονή ανά τύπο λεωφορείου και την συνολική κίνηση όσον αφορά επιβάτες και λεωφορεία που έχει παρατηρηθεί μέχρι στιγμής στο σταθμό.

Η διεργασία mystation αναλαμβάνει το διάβασμα από ένα configuration file που περιέχει βασικές παραμέτρους για τη λειτουργία του σταθμού όπως:

1. αριθμός νησίδων στάθμευσης,
2. χωρητικότητα για στάθμευση ανά τύπο νησίδα,
3. μέγιστος αριθμός επιβατών που μπορεί να έχει κάθε λεωφορείο,
4. μέγιστος χρόνος στάθμευσης οχημάτων.

Στη συνέχεια αναλαμβάνει τη δημιουργία κοινής μνήμης (shared memory) όπου και παραμένει το reference ledger με όλες τις χρήσιμες πληροφορίες στις οποίες έχουν πρόσβαση όλες οι διεργασίες. Επίσης το mystation μπορεί να αναλάβει και τη δημιουργία όλων των διεργασιών. Η κοινή μνήμη θα πρέπει να έχει πληροφορίες για τη παρούσα κατάσταση αλλά και το παρελθόν στο σταθμό. Επίσης, είναι ο χώρος μνήμης από τον οποίο μπορούν να εξαχθούν στατιστικά που ενδιαφέρουν.

Σχεδιασμός των Προγραμμάτων σας:

Έχετε ελευθερία να επιλέξετε οποιαδήποτε δομή επιθυμείτε για τα προγράμματα σας. Θα πρέπει να υιοθετήσετε ένα (περιορισμένο) αριθμό από σηματοφόρους, βοηθητικές δομές και 3-4 ειδών προγράμματα. Όπως έχουμε αναφέρει, ίσως θα ήταν καλή ιδέα να αναπτύξετε ένα πρόγραμμα (το mystation) που δημιουργεί το shared segment, αρχικοποιεί δομές και καταστάσεις, και τέλος γνωστοποιεί το ID του κοινού τμήματος σε άλλα ενδιαφερόμενα εκτελέσιμα από την γραμμή εντολής τους. Το εν λόγω πρόγραμμα θα μπορούσε να αρχικοποιήσει και τους σηματοφόρους που είναι απαραίτητοι για την λύση που επιθυμούμε.

Τα προγράμματά σας θα πρέπει να δημιουργούν εξόδους που με εύκολα κατανοητό τρόπο να μπορούν να δείξουν την ορθότητα αλλά και το ταυτόχρονο της εκτέλεσής τους με άλλες διεργασίες.

Όταν τα προγράμματα σας τερματίζουν, επιβάλλεται να γίνει *καθαρισμός και διαγραφή* του shared segment και των σηματοφόρων που χρησιμοποιήθηκαν. Η απελευθέρωση (purging) τέτοιων πόρων είναι επιτακτική. Σε διαφορετική περίπτωση υπάρχει κίνδυνος ο πυρήνας να μην μπορεί να εξυπηρετήσει μέλλουσες ανάγκες.

Γραμμή Κλήσης των Προγραμμάτων:

Το πρόγραμμα bus μπορεί να κληθεί ως εξής:

```
./bus -t type -n incpassengers -c capacity -p parkperiod -m mantime -s shmid  
όπου
```

- bus είναι το εκτελέσιμο του οχήματος,
- η σημαία -t type παρέχει τον τύπο ταξιδιού του λεωφορείου: ASK, PEL και VOR.
- η σημαία -n παρέχει τον αριθμό επιβατών που μεταφέρει το όχημα,

- η σημαία `-c` παρέχει το μέγιστο αριθμό ατόμων που μπορούν να επιβιβαστούν στο επόμενο ταξίδι,
- η σημαία `-p parkperiod` παρέχει την μέγιστη δυνατή χρονική διάρκεια για την οποία το όχημα θα παραμεινεί παρκαρισμένο. Αυτή η αναμονή μπορεί να υλοποιηθεί κάνοντας `sleep()`. Μέσα σε αυτό το διάστημα 'μη-δραστηριότητας'/`sleep()` στο σταθμό, το λεωφορείο μπορεί να καταθέσει πληροφορίες για το κοινό που μόλις εξυπηρέτησε και οποιαδήποτε άλλη πληροφορία στο ledger.
- η σημαία `-m mantime` παρέχει την χρονική διάρκεια που έχει το όχημα στην διάθεσή του για να κάνει την μανούβρα στάθμευσης/αναχώρησης μόλις έχει πάρει το `ok` από τον `station-manager`.
- η σημαία `-s shmid` δίνει το κλειδί που έχει το κοινό τμήμα μνήμης (και όπου βρίσκονται δομές, σηματοφόροι, και οποιαδήποτε άλλη βοηθητική δομή/μεταβλητή που απαιτείται).

Το πρόγραμμα `station-manager` μπορεί να κληθεί εξής:

```
./station-manager -s shmid
```

όπου

- `station-manager` είναι το εκτελέσιμο που κάνει την διαχείριση του σταθμού,
- η σημαία `-s shmid` δίνει το κλειδί που έχει το κοινό τμήμα μνήμης (και όπου βρίσκονται σηματοφόροι και οποιαδήποτε άλλη βοηθητική δομή/μεταβλητή που απαιτείται).

Το πρόγραμμα επιτηρητής μπορεί να καλεστεί ως εξής:

```
./comptroller -d time -t stattimes -s shmid
```

όπου

- `comptroller` είναι το εκτελέσιμο,
- η σημαία `-d time` παρέχει την χρονική διάρκεια στο τέλος της οποίας η `comptroller` δίνει την κατάσταση που επικρατεί στο σταθμό την εν λόγω χρονική στιγμή.
- η σημαία `-t stattimes` παρέχει την χρονική διάρκεια στο τέλος της οποίας ο `monitor` υπολογίζει και παρέχει στατιστικά για την χρήση της υποδομής.
- η σημαία `-s shmid` δίνει το κλειδί που έχει το κοινό τμήμα μνήμης (και όπου μπορεί να βρίσκονται σηματοφόροι, και οποιαδήποτε άλλη βοηθητική δομή/μεταβλητή που απαιτείται).

Το πρόγραμμα `myport` τότε μπορεί να κληθεί ως εξής:

```
./mystation -l configfile
```

όπου

- `mystation` είναι το εκτελέσιμο,
- η σημαία `-l` ορίζει το αρχείο `configfile` με πληροφορίες για οτιδήποτε έχει με την διαρρύθμισή του σταθμού αλλά και την μέγιστη ώρα διαμονής των οχημάτων.

Η σειρά με την οποία εμφανίζονται οι σημαίες δεν είναι προκαθορισμένη. Προφανώς μπορείτε να χρησιμοποιήσετε πιο πολλές σημαίες στα παραπάνω προγράμματα ώστε να διευκολυνθείτε ή και να σχεδιάσετε τα προγράμματά σας *ριζικά διαφορετικά*.

Τέλος καλό θα ήταν –και για λόγους επιβεβαίωσης και φυσικά μόνο για το σκοπό της άσκησης– να υπάρχει ένα μηχανισμός logging με την μορφή ενός append-only αρχείου. Εδώ, αποτυπώνεται όλη η δραστηριότητά μέχρι στιγμής ώστε να έχετε ένα εύκολο τρόπο να βλέπετε τι γίνεται *όσον αφορά στην ταυτόχρονη εκτέλεση των παραπάνω προγραμμάτων*.

Τι πρέπει να Παραδοθεί:

1. Μια σύντομη και περιεκτική εξήγηση για τις επιλογές που έχετε κάνει στο σχεδιασμό του προγράμματος σας (2-3 σελίδες σε ASCII κειμένου είναι αρκετές).
2. Οποσδήποτε ένα Makefile (που να μπορεί να χρησιμοποιηθεί για να γίνει αυτόματα το compile του προγράμματος σας). Πιο πολλές λεπτομέρειες για το (Makefile) και πως αυτό δημιουργείται δίνονται στην ιστοσελίδα του μαθήματος.
3. Ένα tar-file με όλη σας την δουλειά σε έναν κατάλογο που πιθανώς να φέρει το όνομα σας και θα περιέχει όλη σας την δουλειά δηλ. source files, header files, output files (αν υπάρχουν) και οτιδήποτε άλλο χρειάζεται.

Άλλες Σημαντικές Παρατηρήσεις:

1. Οι εργασίες είναι ατομικές.
2. Το πρόγραμμα σας θα πρέπει να τρέχει στα Linux συστήματα του τμήματος αλλιώς δεν μπορεί να βαθμολογηθεί.
3. Αν και αναμένεται να συζητήσετε με φίλους και συνεργάτες το πως θα επιχειρήσετε να δώσετε λύση στο πρόβλημα, αντιγραφή κώδικα (οποιαδήποτε μορφής) είναι κάτι που δεν επιτρέπεται και δεν πρέπει να γίνει. Οποιοσδήποτε βρει/εί αναμειγμένος σε αντιγραφή κώδικά απλά παίρνει μηδέν στο μάθημα. Αυτό ισχύει για όσους εμπλέκονται ανεξάρτητα από το ποιος έδωσε/πήρε κλπ.
4. Το παραπάνω ισχύει αν διαπιστωθεί *έστω και μερική άγνοια* του κώδικα που έχετε υποβάλει ή άπλα υπάρχει υποψία ότι ο κώδικας είναι προϊόν συναλλαγής με τρίτο/-α άτομο/α.
5. Προγράμματα που δεν χρησιμοποιούν separate compilation χάνουν αυτόματα 5% του βαθμού.
6. Σε καμιά περίπτωση τα Windows δεν είναι επιτρεπτή πλατφόρμα για την παρουσίαση αυτής της άσκησης.