

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
3η Εργασία - Τμήμα: Περιττών Αριθμών Μητρώου
Κ22: Λειτουργικά Συστήματα – Χειμερινό Εξάμηνο '20
Ημερομηνία Ανακοίνωσης: Παρασκευή 20 Νοεμβρίου 2020
Ημερομηνία Υποβολής: Πέμπτη 17 Δεκεμβρίου 2020 Ώρα 23:59

Εισαγωγή στην Εργασία:

Ο στόχος της εργασίας είναι να δημιουργήσετε ανεξάρτητα προγράμματα τα οποία *τρέχουν ταυτόχρονα* και προσομοιώνουν την λειτουργία της ετοιμασίας σαλάτας για λογαριασμό της αλυσίδας η «Γρήγορη Σαλάτα». Έτσι πρέπει να υλοποιήσετε τουλάχιστον 2 *τύπους διεργασιών* (templates) που η κάθε μία έχει διαφορετικό ρόλο. Οι 2 τύποι διεργασιών είναι:

1. Ο Διευθυντής Σαλάτας (chef) συντονίζει και διευθύνει την παραγωγή σαλατών με την βοήθεια υπαλλήλων που ονομάζονται σαλατοποιοί.
2. Οι σαλατοποιοί (saladmakers) αναμένουν να τους παραδοθούν μέρος των υλικών ώστε να προχωρήσουν στην δημιουργία και σύνθεση των σαλατών.

Στην εργασία αυτή, θα πρέπει να υλοποιήσετε 1 διευθυντή και 3 σαλατοποιούς οι οποίοι θα δουλεύουν *ταυτόχρονα* ώστε να δημιουργήσουν ένα συγκεκριμένο αριθμό από σαλάτες που αποτελούνται από *ντομάτες, πιπεριες και κρεμμύδια*.

Στην άσκηση θα πρέπει να:

1. χρησιμοποιήσετε από σηματοφόρους ώστε να έχετε μια επιτυχή συνεργασία μεταξύ των ανεξάρτητων διεργασιών, *shared memory segment* για την σωστή εξυπηρέτηση των εμπλεκόμενων, όπως επίσης και να δημιουργήσετε σχετικές δομές όπου νομίζετε ότι αυτές χρειάζονται,
2. έχετε όλες τις διεργασίες να προσαρτούν (δηλ., *attach*) το παραπάνω *shared memory segment* ώστε να μπορούν να προσπελάσουν μεταβλητές/δομές ενδιαφέροντός τους, και
3. χρησιμοποιήστε POSIX Semaphores για να υλοποιήσετε τη λύση σας που θα εμπεριέχει σχετικές κλήσεις *P()* και *V()*, για εργασίες που διαρκούν 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 με τρόπο που είναι άμεσα κατανοητός/μετρήσιμος.

Διαδικαστικά:

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

Ο πηγαίος κώδικας σας (source code) πρέπει να αποτελείται από *τουλάχιστον δυο* (και κατά προτίμηση πιο πολλά) διαφορετικά αρχεία και θα πρέπει *απαραιτήτως να γίνεται χρήση separate compilation*.

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/fall2020/k22/home>

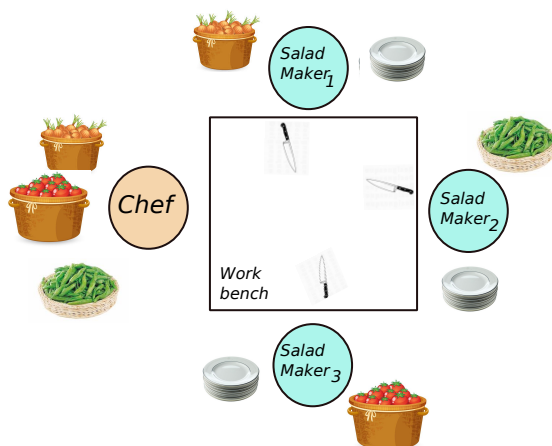
- Υπεύθυνοι για την άσκηση αυτή (ερωτήσεις, αξιολόγηση, βαθμολόγηση κλπ.) είναι ο κ. Μιχάλης Κωνσταντόπουλος *mkonstan+AT-di*, και ο κ. Γιώργος Παναγιωτόπουλος *cs2190012+AT-di*.

Η Σύνθεση των Ταυτοχρόνων Διεργασιών:

Το Σχήμα 1 δείχνει μια αντιπροσωπευτική εικόνα για το πρόβλημα που επιχειρούμε να λύσουμε με ταυτόχρονες διεργασίες: να παρασκευαστεί ένα συγκεκριμένος αριθμός από σαλάτες που η κάθε μία αποτελείται από 1) 1 ντομάτα, 2) 1 πράσινη πιπεριά, και 3) 1 μικρό κρεμμύδι. Για την δημιουργία των σαλατών δουλεύει μια ομάδα από 4 διεργασίες μια εκ των οποίων είναι ο «διευθυντής» (chef) και οι άλλες 3 οι «σαλατοποιιοι» (salad-makers).

Ο διευθυντής έχει στην διάθεσή του και τα 3 παραπάνω υλικά σε κάδους που βρίσκονται πίσω η δίπλα του και είναι συνεχώς τροφοδοτούμενοι (απλό τρίτους εργαζόμενους). Ο κάθε σαλατοποιός έχει στην διάθεσή του μόνο ένα υλικό σε ένα καλάθι που είναι δίπλα του που είναι και αυτό συνεχώς τροφοδοτούμενο. Έτσι ένας σαλατοποιός αναμένει να πάρει τα 2 που του λείπουν ώστε να προχωρήσει στην παρασκευή της σαλάτας, μια εργασία που διαρκεί για ένα τυχαίο χρονικό διάστημα. Μόλις κόψει όλα τα υλικά στο τυχαίο αυτό διάστημα, τοποθετεί τα κομμάτια σε ένα πιάτο και τα δίνει σε έναν σερβιτόρο (ένας είναι πάντα διαθέσιμος).

Το Σχήμα 1 δίνει μια συνολική εικόνα του τι διαμείβεται μεταξύ του διευθυντή και των σαλατοποιών. Οι



Σχήμα 1: Η ταυτόχρονη παραγωγή σαλάτας για την αλυσίδα η «Γρήγορη Σαλάτα».

υποθέσεις/περιορισμοί για τις 4 ταυτόχρονες διεργασίες είναι:

1. Τα λαχανικά στα καλάθια όπως και τα πιάτα για τους σαλατοποιούς είναι πάντα διαθέσιμα.
2. Ο κάθε σαλατοποιός χρησιμοποιεί το υλικό στο οποίο ο ίδιος έχει άμεση προσπέλαση και περιμένει να έλθουν στο τραπέζι τα υλικά που δεν έχει (για να τα πάρει). Για παράδειγμα, ο σαλατοποιός 1 περιμένει να 'ειδοποιηθεί' για την άφιξη στο τραπέζι εργασίας της 'ντομάτας και της πιπεριάς'.
3. Όταν ένας σαλατοποιός έχει όλα τα υλικά στην διάθεσή του, αρχίζει να τα κόβει ώστε να δημιουργήσει μια σαλάτα. Αυτή η διαδικασία παίρνει κάποια ώρα και όταν ολοκληρωθεί, ο σαλατοποιός παραδίδει το πιάτο στο πάντα διαθέσιμο σερβιτόρο.
4. Ο διευθυντής έχει στην διάθεσή του 3 καλάθια και οποιαδήποτε φορά παίρνει τυχαία υλικά από 2 καλάθια και τα βάζει στην τραπέζι (έχει μόνο 2 χέρια!). Ο εκάστοτε συνδυασμός λαχανικών μπορεί να εφοδιάσει έναν συγκεκριμένο σαλατοποιό την φορά. Ο διευθυντής 'ειδοποιεί' τον σαλατοποιό να πάρει τα υλικά που χρειάζεται. Για παράδειγμα αν επιλέξει ντομάτα και κρεμμύδι τότε ειδοποιεί τον σαλατοποιό 2. Μόλις τα υλικά παραλειφθούν από το σαλατοποιό ο διευθυντής συνεχίζει στο ίδιο μοτίβο μέχρι να ολοκληρωθεί η προετοιμασία των σαλατών που πρέπει να γίνει.

Το Ζητούμενο & Αποτελέσματα

Τα ανεξάρτητα προγράμματα για τον διευθυντή και τους πελάτες πρέπει να συγχρονίζονται με POSIX P()/V()

και να έχουν προσπέλαση σε οποιαδήποτε δομή/μεταβλητή που χρειάζονται στη κοινή μνήμη (shared memory).

Έχετε ελευθερία να υιοθετήσετε ένα (περιορισμένο) αριθμό από σηματοφόρους, και βοηθητικές μεταβλητές/δομές για τα προγράμματα σας. Θα ήταν καλή ιδέα το πρόγραμμα `chef` να δημιουργεί το shared segment, αρχειοποιεί δομές ή/και μεταβλητές. Το ID του κοινού τμήματος μπορεί να γίνει γνωστό στα εκτελέσιμα των `saladmakers` από την γραμμή εντολής των τελευταίων. Το πρόγραμμα `chef` θα μπορούσε να αρχικοποιήσει και τους σηματοφόρους που είναι απαραίτητοι για την λύση που επιθυμούμε.

Στο shared segment αναμένεται να υπάρχει μια μεταβλητή η οποία σε οποιαδήποτε χρονική στιγμή αναφέρει τον αριθμό των σαλατών που θα πρέπει ακόμα να παρασκευαστούν για να λήξει η συνεργασία. Οι `saladmakers` ολοκληρώνουν όταν η λόγω μεταβλητή έχει την τιμή 0.

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 των εμπλεκόμενων διεργασιών.

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 της Εφαρμογής:

Η εφαρμογή `chef` μπορεί να κληθεί με τον παρακάτω τρόπο στην γραμμή εντολής (tty):

```
./chef -n numOfSlds -m mantime
```

όπου

- `chef` είναι το εκτελέσιμο που θα δημιουργήσετε,
- η σημαία `-n numOfSlds` παρέχει τον αριθμό των σαλατών που πρέπει να παρασκευαστούν, και
- η σημαία `-m mantime` παρέχει την χρονική διάρκεια που έχει στην διάθεση του ο διευθυντής να πάρει μια 'ανάσα' μεταξύ διαδοχικών εναποθέσεων υλικών στο τραπέζι.

Το πρόγραμμα `saladmaker` μπορεί να κληθεί εξής:

```
./saladmaker -t1 lb -t2 ub -s shmid
```

όπου

- `saladmaker` είναι το εκτελέσιμο που υλοποιεί την λειτουργία του σαλατοποιού,
- οι σημαίες `-t1` και `-t2` προσδιορίζουν το χρονικό διάστημα που ο σαλατοποιός χρειάζεται για να κάνει την δουλειά του. Έτσι, επιλέγεται από το πρόγραμμα τυχαία μια χρονική διάρκεια μεταξύ των ορίων `lb` και `ub`), και,
- 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).

Η σειρά με την οποία εμφανίζονται οι σημαίες δεν είναι προκαθορισμένη. Προφανώς μπορείτε να χρησιμοποιήσετε πιο πολλές σημαίες στα παραπάνω προγράμματα ώστε να διευκολυνθείτε.

Τέλος καλό θα ήταν –και για λόγους επιβεβαίωσης– να υπάρχει ένα μηχανισμός logging με την μορφή ενός append-only αρχείου για την δουλειά που όλοι επιτελούν. Εδώ, αποτυπώνεται όλη η δραστηριότητα της κάθε διεργασίας μέχρι στιγμής ώστε να έχετε ένα εύκολο τρόπο να βλέπετε τι γίνεται συνολικά *όσον αφορά στην ταυτόχρονη εκτέλεση των προγραμμάτων*.

Αποτελέσματα των Διεργασιών:

Ο `chef` πρέπει κατ ελάχιστον να δίνει τα εξής στοιχεία:

1. Αριθμός σαλατών που έχουν κατασκευαστεί συνολικά.

2. Ο αριθμός σαλατών που ο κάθε σαλατοποιός έχει φτιάξει.
3. Χρονικά logs που για κάθε σαλατοποιό δίνουν το timeline της δραστηριότητας του (1 τέτοιο αρχείο για κάθε σαλατοποιό).
4. Λίστα από χρονικά διαστήματα στα οποία 2 η πιο πολλές διεργασίες δρουν ταυτόχρονα.

Πιο λεπτομερή περιγραφή της εξόδου, δίνεται στην σελίδα του μαθήματος.

Περιορισμοί για τα Προγράμματα που θα Γράψετε:

1. Δεν μπορείτε να κάνετε pre-allocate οποιοδήποτε χώρο αφού δομή(-ές) θα πρέπει να μπορεί(-ουν) να μεγαλώσει(-ουν) χωρίς ουσιαστικά κανέναν περιορισμό όσον αφορά στον αριθμό των εγγράφων που μπορούν να αποθηκεύσουν. Η χρήση στατικών πινάκων/δομών που δεσμεύονται στην διάρκεια της συμβολομετάφρασης του προγράμματος σας δεν είναι αποδεκτές επιλογές.
2. Για οποιαδήποτε σχεδιαστική επιλογή που θα υιοθετήσετε, θα πρέπει να να την δικαιολογήσετε στην αναφορά σας.
3. Πριν να τερματίσει η εκτέλεση των προγραμμάτων το περιεχόμενο των δομών απελευθερώνεται με σταδιακό και ελεγχόμενο τρόπο. Εξίσου σημαντικό είναι η κώδικας για τον chef πριν τερματίσει να καθαρίζει πλήρως τμήματα της shared memory ώστε ο εν λόγω πόρος να αποδοθεί σωστά πίσω στον πυρήνα.
4. Αν πιθανώς κάποια κομμάτια του κωδικά σας προέλθουν από κάποια δημόσια πηγή, θα πρέπει να δώσετε αναφορά στη εν λόγω πηγή είτε αυτή είναι βιβλίο, σημειώσεις, Internet URL κλπ. και να εξηγήσετε πως ακριβώς χρησιμοποιήσατε την εν λόγω αναφορά.
5. Έχετε πλήρη ελευθερία να επιλέξετε το τρόπο με τον οποίο τελικά θα υλοποιήσετε βοηθητικές δομές.

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 η αναφορά σας.

Άλλες Σημαντικές Παρατηρήσεις:

1. Οι εργασίες είναι ατομικές.
2. Το πρόγραμμά 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 βρεθεί αναμειγμένος σε αντιγραφή κώδικα 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 επιλέξιμη πλατφόρμα για την παρουσίαση αυτής της άσκησης.