ΚΑΡΑΒΑΓΓΕΛΗ ΕΥΤΥΧΙΑ
ΑΜ: 1115201800062

ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΕΡΓΑΣΙΑ 1

Τα παραδοθέντα αρχεία είναι τα εξής: p1.c, p2.c, ENC1.c, ENC2.c, channel.c, semoper.h και semoper.c. Περιλαμβάνεται επιπλέον Makefile, οπότε για την μεταγλώττιση του προγράμματος οι εντολές είναι: ''make clean'' και ''make all''. (Πριν την εκτέλεση του προγράμματος καλό θα ήταν να εκτελεστεί η εντολή ''ipcrm -a'' για να αποδεσμευτούν τυχόν σημαφόροι και τμήματα διαχειριζόμενης μνήμης. Για την εκτέλεση του προγράμματος να χρησιμοποιηθούν δύο terminals. Στο 1ο να γραφτεί ''./ p1'' και στο 2ο ''./p2''. Σημαντικό είναι να δοθεί πρώτα η εντολή για να εκτελεστεί το εκτελέσιμο ./p1 και μετά για το ./p2, διότι η p1 διεργασία παράγει το κλειδί για τη δημιουργία των σημαφόρων και της διαχειριζόμενης μνήμης που χρησιμοποιούν όλες οι διεργασίες. Η γλώσσα υλοποίησης είναι C. Έγινε δοκιμή στον linux04 και δεν παρατηρήθηκε κάποιο θέμα.

semoper.h

Περιλαμβάνει βιβλιοθήκες για τους σημαφόρους και τη διαχειριζόμενη μνήμη. Ορίζεται ακόμα το μέγεθος της διαχειριζόμενης μνήμης που θα δημιουργηθεί και το μέγιστο μήκος μηνύματος που μπορεί να δώσει ο χρήστης (μέχρι 512 χαρακτήρες). Τα δεδομένα που βρίσκονται στη διαχειριζόμενη μνήμη έχουν τη μορφή ενός struct το οποίο περιέχει το μήνυμα που στέλνει ο εκάστοτε χρήστης καθώς και μια μεταβλητή που δηλώνει από ποιον χρήστη γράφτηκε το μήνυμα που θα βρίσκεται κάθε φορά στη διαχειριζόμενη μνήμη. Αν είναι 1, το μήνυμα γράφτηκε από το χρήστη της p1 διεργασίας διαφορετικά αν είναι 0 το μήνυμα γράφτηκε από το χρήστη της διεργασίας p2. Δίνονται ακόμα τα πρωτότυπα συναρτήσεων που χρησιμοποιούνται και από τις 5 διεργασίες, δηλαδή τα αρχεία p1.c, p2.c, ENC1.c, ENC2.c, channel.c.

semoper.c

Το αρχείο αυτό περιέχει συναρτήσεις για την αρχικοποίηση της δομής των σημαφόρων καθώς και τον έλεγχο των σημαφόρων που χρησιμοποιούν όλες οι διεργασίες (δηλαδή όλα τα .c αρχεία), δηλαδή σε τι τιμές θα αρχικοποιηθούν οι σημαφόροι από τη διεργασία p1, που είναι αυτή που ξεκινάει και την επικοινωνία και ποιος σημαφόρος θα γίνεται κάθε φορά DOWN και UP ανάλογα με το ποια διεργασία εκτελεί το critical section της. Η δομή των σημαφόρων που δημιουργείται, αποτελείται από 6 σημαφόρους, εκ των οποίων αυτός με index 0, αρχικοποιείται στο 1 και όλοι οι άλλοι στο 0 (θα πρέπει να ελευθερωθούν για να εκτελέσει η κάθε διεργασία το critical section της).

Λειτουργία σημαφόρων ανάλογα με το index τους:

0: όταν είναι δεσμευμένος η διεργασία p1 εκτελεί το critical section της
1: όταν είναι δεσμευμένος η διεργασία enc1 εκτελεί το critical section της

2: απελευθερώνεται από την p2 για να μπορέσει η p1 να διαγράψει τη δομή των σημαφόρων και τη διαχειριζόμενη μνήμη αφότου όλες οι υπόλοιπες διεργασίες έχουν αποδεσμεύσει αυτές τις δομές.
3: όταν είναι δεσμευμένος η διεργασία enc2 εκτελεί το critical section της
4: όταν είναι δεσμευμένος η διεργασία p2 εκτελεί το critical section της
5: όταν είναι δεσμευμένος η διεργασία channel εκτελεί το critical section της

Ακόμα υπάρχει η συνάρτηση "get_key()" που την καλούν όλες οι διεργασίες για να έχουν το ίδιο κλειδί ώστε να αποκτήσουν τους ίδιους σημαφόρους και το ίδιο κομμάτι διαχειριζόμενης μνήμης. Το κλειδί αυτό δημιουργείται από το αρχείο p1.c και τον αριθμό 1, οπότε είναι σημαντικό η διεργασία p1, δηλαδή το αρχείο p1.c, να εκτελεστεί πρώτο.

p1.c

Προσομοιώνει τη λειτουργία του χρήστη p1 που ξεκινάει την επικοινωνία. Δημιουργεί μια δομή με 6 σημαφόρους καθώς και τη διαχειριζόμενη μνήμη, ώστε να μπορέσουν στη συνέχεια οι υπόλοιπες διεργασίες να τα αποκτήσουν. Για τη διαχειριζόμενη μνήμη υπάρχουν δύο pointers (p_shared_data και ptr1), οι οποίοι δείχνουν σε δύο διαφορετικά σημεία της διαχειριζόμενης μνήμης. Στο ένα κομμάτι θα υπάρχει πάντα το μήνυμα που έδωσε ο χρήστης (p1 user, p2 user) και στο άλλο κομμάτι θα γίνονται αλλαγές πάνω στα δεδομένα (δηλαδή θα αλλάζεται το μήνυμα, θα βρίσκεται το checksum κτλ). Ταυτόχρονα, εκκινείται και η διεργασία enc1, η οποία προσομοιώνεται στο ENC1.c αρχείο, η οποία και θα αφεθεί να εκκινήσει μόνο όταν κάποιος χρήστης δώσει μήνυμα. Ο χρήστης p1, δίνει μήνυμα από το πληκτρολόγιο και αυτό αντιγράφεται στο κομμάτι της διαχειριζόμενης μνήμης που δείχνει ο δείκτης p_shared_data, και στη συνέχεια ελευθερώνεται ο σημαφόρος της enc1 διεργασίας, ώστε αυτή να εκκινήσει. Κάθε φορά που είναι η σειρά του χρήστη p1 να στείλει μήνυμα εκτυπώνει και το μήνυμα που έλαβε από τον χρήστη p2, πριν στείλει το δικό του, εκτός από την 1η φορά αφού ο p1 είναι αυτός που εκκινεί την επικοινωνία.

Όταν κάποιος από τους δύο χρήστης δώσει το μήνυμα τερματισμού "TERM", ο p1 είναι υπεύθυνος για την διαγραφή των σημαφόρων και της διαχειριζόμενης μνήμης, αφού ήταν εκείνος που τα δημιούργησε, οπότε περιμένει μέχρι όλες οι υπόλοιπες διεργασίες να αποδεσμεύσουν τις δομές αυτές. Ενημερώνεται από την p2 για αυτό το σκοπό, μόλις μπορέσει να κάνει DOWN το σημαφόρο με index 6.

ENC1.c

Προσομοιώνεται η λειτουργία της διεργασίας enc1 της εκφώνησης. Αφού αποκτήσει με επιτυχία τους σημαφόρους και τη διαχειριζόμενη μνήμη που δημιουργήθηκαν από την p1, εκκινεί και τη διεργασία channel, η οποία και θα αφεθεί να εκτελέσει μόνο όταν η enc1 ολοκληρώσει το critical section της, δηλαδή να βρει το checksum του αρχικού μηνύματος και να φέρει στη διαχειριζόμενη μνήμη το αρχικό μήνυμα μαζί με το αρχικό checksum. Οπότε η enc1, βρίσκει το checksum από το μήνυμα του χρήστη (με το md5) και αντιγράφει το μήνυμα του χρήστη μαζί με το checksum του στο κομμάτι της διαχειριζόμενης μνήμης στο οποίο δείχνει ο δείκτης ptr1. Στο κομμάτι της

διαχειριζόμενης μνήμης στο οποίο δείχνει ο δείκτης p_shared_data εξακολουθεί να υπάρχει το αρχικό μήνυμα που έδωσε ο χρήστης. Κατόπιν ενημερώνεται η διεργασία channel να εκκινήσει.

channel.c

Προσομοιώνεται η λειτουργία της διεργασίας channel της εκφώνησης. Αφού αποκτήσει με επιτυχία τους σημαφόρους και τη διαχειριζόμενη μνήμη που δημιουργήθηκαν από την p1, αποφασίζεται με πιθανότητα 0.2 αν το αρχικό μήνυμα του χρήστη θα αλλαχθεί ή όχι. Μέσω της rand() παράγονται τυχαίοι αριθμοί από το 0 έως το 1. Αν ο αιρθμός που θα παραχθεί είναι μικρότερος από 0.2 τότε το μήνυμα θα υποστεί αλλοίωση. Διαφορετικά θα παραμείνει το ίδιο. Σε περίπτωση που αλλαχθεί το μήνυμα, αυτό που συμβαίνει είναι ότι στη διαχειριζόμενη μνήμη στο κομμάτι που βρίσκεται το αρχικό μήνυμα μαζί με το checksum (δηλαδή στο κομμάτι που δείχνει ο ptr1), επί 5 φορές σε κάποιες από τις θέσεις που βρίσκονται οι χαρακτήρες του μηνύματος του χρήστη θα τοποθετηθούν νέοι τυχαίοι χαρακτήρες. Φυσικά στο κομμάτι που δείχνει ο p_shared_data εξακολουθεί να υπάρχει το αρχικό μήνυμα του χρήστη. Μόλις ολοκληρωθεί η διαδικασία ενημερώνεται η διεργασία enc2 να εκκινήσει.

ENC2.c

Προσομοιώνεται η λειτουργία της διεργασίας enc1 της εκφώνησης. Αφού αποκτήσει με επιτυχία τους σημαφόρους και τη διαχειριζόμενη μνήμη που δημιουργήθηκαν από την p1, επιχειρεί να βρει το checksum του νέου μηνύματος (το οποίο μπορεί να άλλαξε μπορεί και όχι). Αντιγράφει το μήνυμα από το shared memory σε ένα νέο χώρο για να βρει το νέο checksum του και αντιγράφει και το παλιό checksum από το shared memory σε έναν άλλο χώρο. Βρίσκει το νέο checksum και μετά κάνει σύγκριση μεταξύ των 2 checksums. Αν είναι ίδια τότε εκκινείται η διεργασία για τον χρήστη p2 (αν ήταν ο p1 εκείνος που είχε στείλει το μήνυμα), ώστε να εμφανίσει το μήνυμα του p1 και να στείλει το δικό του. Αλλιώς, εκκινείται η διεργασία για τον χρήστη p1 (αν ήταν ο p2 εκείνος που είχε στείλει το μήνυμα), ώστε να εμφανίσει το μήνυμα του p2 και να στείλει το δικό του. Και η διαδικασία επαναλαμβάνεται. Αν τα δύο checksums δεν είναι ίδια, αυτό σημαίνει ότι το μήνυμα υπέστη αλλοίωση, οπότε εκκινείται η διεργασία enc1 ώστε να εκτελεστεί ξανά η διαδικασία. Σε περίπτωση που κάποιος χρήστης δεν λάβει το μήνυμα την 1η φορά, θα εμφανιστεί μήνυμα στην οθόνη και θα εκτελεστεί η κρίσιμη περιοχή της enc1 πάλι. Επειδή η πιθανότητα αλλαγής του μηνύματος είναι σχετικά μικρή, τη 2η ή την 3η φορά ο χρήστης θα λάβει το μήνυμα, ώστε να στείλει μετά το δικό του.

p2.c

Προσομοιώνει τη λειτουργία του χρήστη p2 που αρχικά περιμένει μήνυμα από τον p1 για να στείλει το δικό του. Αφού αποκτήσει με επιτυχία τους σημαφόρους και τη διαχειριζόμενη μνήμη που δημιουργήθηκαν από την p1, εκκινεί και τη διεργασία enc2, η οποία προσομοιώνεται στο ENC2.c αρχείο, η οποία και θα αφεθεί να εκκινήσει μόνο όταν ενημερωθεί από το κανάλι. Επειδή, αρχικά ο p2 περιμένει μήνυμα από τον p1 για να

στείλει το δικό του παραμένει μπλοκαρισμένος μέχρι να ενημερωθεί από την διεργασία enc2. Μόλις λάβει το μήνυμα το τυπώνει και δίνει το δικό του προς αποστολή στον p1, το οποίο αντιγράφεται στο κομμάτι της διαχειριζόμενης μνήμης που δείχνει ο δείκτης p_shared_data, και στη συνέχεια ελευθερώνεται ο σημαφόρος της enc1 διεργασίας, ώστε αυτή να εκκινήσει. Οι κώδικες των χρηστών p1 και p2 είναι συμμετρικοί.

ΣΧΟΛΙΑ:

Το critical section της κάθε διεργασίας βρίσκεται μέσα σε while loop το οποίο και σταματάει μόλις η κάθε διεργασία ξεχωριστά αντιληφθεί ότι κάποιος από τους δύο χρήστες έδωσε το μήνυμα τερματισμού "TERM". Το μήνυμα ''TERM'' δεν υφίσταται ποτέ αλλοίωση από το κανάλι και επιπλέον οποιαδήποτε συμβολοσειρά δοθεί που οι 4 πρώτοι χαρακτήρες της είναι ''TERM'', η επικοινωνία θα διακοπεί.

Σε περίπτωση που φανεί ότι τα μηνύματα λαμβάνονται πάντα σωστά αυτό θα συμβαίνει επειδή θα παράγονται πάντα αριθμοί μεγαλύτεροι από 0.2, οπότε τα μηνύματα δεν θα αλλοιώνεται. Αντίστοιχα αν εμφανίζεται συνέχεια μήνυμα ότι το μήνυμα δεν μπόρεσε να αποσταλεί, σημαίνει ότι παράγονται αριθμοί μικρότεροι του 0.2. Δοκιμάστε αρκετές φορές να στείλετε μηνύματα.

Για την επαναμετάδοση, επειδή ενημερώνεται η διεργασία enc1 να εκκινήσει σε περίπτωση που το μήνυμα αλλοιωθεί, δεν χρειάζεται ο χρήστης να στείλει εκ νέου κάποιο μήνυμα αλλά επειδή αυτό υπάρχει στη διαμοιραζόμενη μνήμη, στο κομμάτι που δείχνει ο δείκτης p_shared_data, στέλνεται ξανά αυτόματα, οπότε ο χρήστης που περιμένει μήνυμα θα το λάβει κάποια στιγμή.

Οι καθυστερήσεις υπάρχουν μέχρι να εμφανιστούν μηνύματα στους χρήστες είναι λόγω του sleep(some_number).