Λειτουργικά Συστήματα

1η Άσκηση

Αναγνωστόπουλος Αθανάσιος

AM: 1115201800006

Αρχικά υπάρχουν 3 αρχεία .c τα p1.c CHAN.c και p2.c και ένα αρχείο arxeio.h στο οποίο υπάρχουν όλες οι βιβλιοθήκες που χρειάζονται καθώς και τα defines.

Στο αρχείο p1.c με fork() δημιουργούνται οι διεργασίες P1 και ENC1 και αντίστοιχα στο αρχείο p2.c οι διαδικασίες P2 και ENC2.

Στο αρχείο CHAN.c υλοποιείται η διεργασία CHAN η οποία αντικαθιστά τους χαρακτήρες του μηνύματος τυχαία με '*' , βάση μιας πιθανότητας που δέχεται ως όρισμα το πρόγραμμα. Αν δεν δοθεί όρισμα η default πιθανότητα είναι 10.

Στο αρχείο p1.c δεσμεύονται 900 bytes shared memory που χρησιμοποιούνται ως εξής:

BUFFER1 [0 – 200]
BUFFER2 [200- 400]
BUFFER3 [400 – 600]
BUFFER4 [600 – 800]
ROI [810]

ENDP1 [820]
ENDENC1 [830]
ENDCHAN [840]
ENDENC2 [850]
ENDP2 [860]

Το BUFFER1 υπάρχει για την επικοινωνία μεταξύ της P1 και ENC1.

Το BUFFER2 υπάρχει για την επικοινωνία μεταξύ της ENC1 και CHAN.

Το BUFFER3 υπάρχει για την επικοινωνία μεταξύ της CHAN και ENC2.

Το BUFFER4 υπάρχει για την επικοινωνία μεταξύ της ENC2 και P2.

Οι διεργασίες ENC1 και ENC2 υπολογίζουν την τιμή κατακερματισμού του μηνύματος (checksum) με τον αλγόριθμο MD5 και δημιουργούν το επαυξημένο μήνυμα ενώνοντας το απλό και το checksum με ένα χαρακτήρα ‘ \0 ’ ενδιάμεσα, έτσι ώστε να γνωρίζουμε που τελειώνει το απλό και που αρχίζει το checksum για να μπορούν να χωριστούν στην συνέχεια στη διεργασία CHAN.

Στη συνέχεια η διεργασία CHAN χωρίζει το απλό μήνυμα από το checksum, αλλάζει μόνο το απλό μήνυμα βάση της πιθανότητας αλλαγής και επανενώνει το αλλαγμένο μήνυμα και το checksum (που δεν έχει αλλάξει) με τον ίδιο τρόπο (προσθέτοντας τον χαρακτήρα ‘ \0 ’ ενδιάμεσα).

Μόλις τελειώσει η CHAN ξαναστέλνει το αλλαγμένο μήνυμα στην διεργασία ENC? (1 ή 2 ανάλογα την ροή) όπου αυτή τη φορά χωρίζουμε το αλλαγμένο μήνυμα (new_message) από το checksum (old_checksum) και υπολογίζουμε το checksum του new_message

με τον αλγόριθμο MD5 (new_checksum). Αν old_checksum == new_checksum προχωράμε στην P? για εκτύπωση του σωστού μηνύματος.

Σε περίπτωση που έχει αλλάξει το μήνυμα στην CHAN (old_checksum != new_checksum) το μήνυμα θα ξανακάνει την διαδρομή ENC? → CHAN → ENC? μέχρι να μην αλλάξει καθόλου και το checksum να είναι το ίδιο πριν και μετά την διεργασία CHAN. Αυτό επιτυγχάνεται κάνοντας up() το semaphore που ξεκινάει το ENC? (sem1 ή sem3 ανάλογα την ροή) για υπολογισμό του checksum ξανά ώστε να επαναληφθεί η διαδρομή ENC? → CHAN → ENC? έως οτου το μήνυμα να είναι σωστό.

Επίσης η μεταβλητή ROI είναι ένα flag που δείχνει την ροή του προγράμματος, δηλαδή όταν το P1 στέλνει στο P2 είναι 0 και όταν το P2 στέλνει στο P1 είναι 1. Αλλάζει κάθε φορά μετά την εκτύπωση του τελικού μηνύματος στην P1 ή στην P2.

Το πρόγραμμα τερματίζει όταν δοθεί ως είσοδος η συμβολοσειρά TERM. Αν στο P1 δώσουμε είσοδο ‘’TERM’’ τότε η διεργασία P1 τερματίζει και κάνει το flag ENDENC1 = 1. Συνεχίζοντας με την σειρά της η ENC1 κάνει την μεταβλητή ENDCHAN = 1 και τερματίζει. Η CHAN κάνει την ENDENC2 = 1 και τερματίζει και τέλος η ENC2 κάνει την ENDP2 = 1 και τερματίζει. Στην P2 αποδεσμεύονται οι δομές συγχρονισμού και τερματίζει και η P2. Αντίστοιχα το ίδιο θα συμβεί και αν δώσουμε είσοδο ‘’TERM’’ στη διεργασία P2.

Semaphores

Υπάρχουν 5 σεμαφόροι sem0,sem1,sem2,sem3 και sem4.

Έστω ότι η P1 στέλνει στην P2.

Μολις τελειώσει η διεργασία P1 κάνει up το σημαφόρο sem1 για να ξεκινήσει η διεργασία ENC1 η οποία περίμενε να τελειώσει η P1.

Μόλις τελειώσει η ENC1 κάνει up τον sem2 έτσι ώστε να ξεκινήσει η CHAN η οποία περίμενε την ENC1.

Αφού τελειώσει η CHAN κάνει up τον sem3 για να ξεκινήσει η ENC2 που περίμενε την CHAN.

Τέλος μόλις τελειώσει η ENC2 ξεκινάει την P2 η οποία περίμενε την ENC2 για την τελική εκτύπωση του μηνύματος.

Αντίστοιχα και όταν η P2 στέλνει στην P1.

Το sem0 χρησιμεύει για την έναρξη της P1 και την εκτύπωση του μηνύματος όταν η P2 στέλνει στην P1 (ROI == 1) και γίνεται up στο τέλος της ENC1.

Μεταγλώττιση και Εκτέλεση

gcc p1.c –o p1 –lcrypto

./p1

gcc CHAN.c –o CHAN

./CHAN 20 (το όρισμα είναι η πιθανότητα αλλαγής)

gcc p2.c –o p2 –lcrypto

./p2