

ΦΟΥΚΑΝΕΛΗΣ ΧΡΗΣΤΟΣ-ΓΕΩΡΓΙΟΣ

1115201900204

ΕΡΓΑΣΙΑ 1 – ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΕΚΕΜΒΡΙΟΣ 2021

DOCUMENTATION / ΕΠΕΞΗΓΗΣΗ ΥΛΟΠΟΙΗΣΗΣ

Η εργασία έχει υλοποιηθεί με system-v σημαφόρους και χρησιμοποιεί κώδικα του φροντιστηρίου ο οποίος έχει τροποποιηθεί κατάλληλα για τις ανάγκες της εργασίας (λαμβάνει υπόψιν παραπάνω από έναν σημαφόρο ενώ ήταν αρχικά υλοποιημένος να λειτουργεί μόνο για έναν). Η διαμοιραζόμενη μνήμη αναπαρίσταται ως μια συμβολοσειρά 100 χαρακτήρων (όσο και μια γραμμή του αρχείου).

Στην εργασία χρησιμοποιείται ένα σύνολο από 3 σημαφόρους. Ο 1^{ος} (με sem_num 0) χρησιμοποιείται για να ελέγχει ποιο από τα παιδιά-διεργασίες θα στείλει το i-οστό (από τα N) αίτημα του στον πατέρα-διεργασία (ελέγχει δηλαδή την πρόσβαση των παιδιών-διεργασιών στην κοινή μνήμη). Αρχικοποιείται σε 1. Ο 2^{ος} (sem_num = 1) σημαφόρος χρησιμοποιείται για να ελέγξει την πρόσβαση του παιδιού διεργασία στην κοινή μνήμη όσο ο server (γονική-διεργασία) εξυπηρετεί το αίτημά του. Ο τελευταίος (sem_num = 2) χρησιμοποιείται για να ελέγξει την πρόσβαση του server στην κοινή μνήμη όσο ο client γράφει το αίτημα του στην κοινή μνήμη (ο server αποκτά πρόσβαση στην κοινή μνήμη μόνο όταν λάβει κάποιο αίτημα και όσο παραμένει ανενεργός δεν έχει πρόσβαση σε αυτήν). Οι δυο τελευταίοι σημαφόροι αρχικοποιούνται σε 0.

Η γονική διεργασία διαβάζει το αρχείο και αποθηκεύει κάθε γραμμή του σε ένα δυσδιάστατο πίνακα (*char **data*) χαρακτήρων (πίνακα συμβολοσειρών) αφού πρώτα δημιουργήσει και αρχικοποιήσει το shared memory segment, το σύνολο των 3 σημαφόρων και κάνει attach την διαμοιραζόμενη μνήμη. Στη συνέχεια δημιουργεί τα K παιδιά-διεργασίες καλώντας την *fork()*. Έπειτα διαχωρίζει τον κώδικα που εκτελούν τα παιδιά με αυτόν της γονικής διεργασίας ο οποίος θα περιγραφεί αναλυτικά παρακάτω. Τέλος, περιμένει όλα τα παιδιά-διεργασίες να τερματίσουν και διαγράφει τους σημαφόρους και την κοινή μνήμη.

PARENT PROCESS

Ο server εκτελεί ένα βρόχο $N \times K$ φορές όσα και τα συνολικά αιτήματα που δημιουργούνται κατά την εκτέλεση του προγράμματος. Η σειρά των αιτημάτων που εξυπηρετεί είναι τυχαία (εξυπηρετεί τη διεργασία η οποία πρόλαβε να λάβει πρόσβαση στο resource του). Αρχικά κάνει down τον 3^ο σημαφόρο (ελέγχοντας έτσι αν έχει λάβει κάποιο αίτημα [όπως αναφέρθηκε και παραπάνω, η γονική διεργασία έχει πρόσβαση στη μνήμη μόνο όταν παιδιά-διεργασίες την «ξυπνήσουν» στέλνοντας της αίτημα]) και στη συνέχεια διαβάζει το αίτημα το οποίο έχει καταγραφεί στην κοινή μνήμη και τοποθετεί σε αυτήν, την γραμμή που αντιστοιχεί στο αίτημα (τη γραμμή αυτή την βρίσκει στον πίνακα *data*). Τέλος κάνει up τον 2^ο σημαφόρο προκειμένου να ξεμπλοκάρει τον πελάτη ώστε να διαβάσει την απάντηση του αιτήματός του.

CHILD PROCESS

Ο client αρχικά κάνει attach την διαμοιραζόμενη μνήμη και έπειτα εκτελεί N επαναλήψεις που στην κάθε μία ακολουθεί την παρακάτω διαδικασία: Αρχικά κάνει down τον 1^ο σημαφόρο (αποπειράται να λάβει πρόσβαση στην κοινή μνήμη ανταγωνιζόμενος τις υπόλοιπες διεργασίες). Θα μπλοκαριστεί μέχρι να καταφέρει να λάβει πρόσβαση στο resource. Στη συνέχεια γράφει το αίτημά του στην κοινή μνήμη με τη χρήση βοηθητικής συνάρτησης (*cast_to_string()*). Η γραμμή την οποία θα αιτηθεί επιλέγεται

με την χρήση της *rand()*. Έπειτα «ξυπνάει» τον server προκειμένου να διαβάσει το αίτημα του κάνοντας up τον 3^ο σημαφόρο και έπειτα κάνει down τον 2^ο σημαφόρο (μπλοκάρεται περιμένοντας τον server να εξυπηρετήσει το αίτημά του). Αφού τελειώσει η γονική διεργασία και διαβάσει την απάντησή της, ελευθερώνει το resource της κοινής μνήμης κάνοντας up τον 1^ο σημαφόρο, δίνοντας τη δυνατότητα στις υπόλοιπες διεργασίες να διεκδικήσουν ποια από όλες θα στείλει αίτημα. Ο πελάτης μετράει το χρόνο που παρέρχεται στο διάστημα που πέρασε ακριβώς αφού έγραψε στην κοινή μνήμη και ακριβώς όταν ξεμπλοκαρίστηκε και έλαβε πρόσβαση στην κοινή μνήμη η οποία περιέχει την απάντηση του αιτήματός του. Ο χρόνος αυτός θεωρείται ότι είναι ο χρόνος τον οποίο η διεργασία περιμένει να εξυπηρετηθεί το αίτημά της βρισκόμενη σε blocked κατάσταση. Η μέτρηση του παρελθόντα χρόνου γίνεται με τη χρήση της *clock()*. Τέλος, και αφού έχουν εξυπηρετηθεί και τα N αιτήματα, εκτυπώνει τον μέσο χρόνο αναμονής και κάνει detach την κοινή μνήμη.

ΣΗΜΕΙΩΣΕΙΣ / ΠΑΡΑΤΗΡΗΣΕΙΣ:

Το πρόγραμμα εκτυπώνει για βοηθητικούς σκοπούς όλα τα αιτήματα που γίνονται (προέλευση, πορεία εξυπηρέτησης κλπ) με αποτέλεσμα να γεμίζουν οι εκτυπώσεις στην έξοδο του προγράμματος για μεγάλα N,K. Αυτό γίνεται ώστε να είναι ξεκάθαρη η λειτουργικότητα του προγράμματος.

Ο κώδικας για τον πελάτη και εξυπηρέτη είναι κοινός και αντί να καλείται η *exec()* για τον πελάτη, η διεργασία παιδί τρέχει τον κώδικα που περιέχει η *clientFunct()*.

Η *rand()* επιστρέφει τον ίδιο αριθμό για μια επανάληψη και στα K παιδιά αλλά αυτό δεν επηρεάζει την λειτουργικότητα του προγράμματος

Ο κατάλογος της εργασίας περιέχει makefile και για να μεταγλωτιστεί και να εκτελεστεί το πρόγραμμα χρειάζεται απλά την εντολή *make run*. Οι παράμετροι εκτέλεσης έχουν οριστεί ενδεικτικά στο makefile όπως και παρέχεται και ενδεικτικό αρχείο (*input.txt*) από το οποίο διαβάζει η μητρική διεργασία.

Ό,τι εκτύπωση αφορά γέννηση / τερματισμό διεργασίας σηματοδοτείται με #, ό,τι εκτύπωση αφορά μέσο χρόνο αναμονής σηματοδοτείται με -> και οι υπόλοιπες απλά αναφέρουν το αναγνωριστικό της διεργασίας. Ο παραδοτέος κώδικας έχει τις εκτυπώσεις σε σχόλιο, και κατά την εκτέλεση εκτυπώνονται τελικά οι μέσοι χρόνοι αναμονής για κάθε διεργασία παιδί.