

READ ME

Νικόλαος Πνευματικός
AM 1115201900157

Εργασία 1:

Αρχικά ανοίγω το αρχείο με την `foren` και διατρέχω κάθε γραμμή του για να βρω πόσες γραμμές έχει συνολικά. Ορίζω έναν πίνακα `strArray` με διαστάσεις συνολικός αριθμός γραμμών \times 100 (όπου 100 οι μέγιστοι χαρακτήρες κάθε γραμμής) και ξανά διατρέχω κάθε γραμμή του αρχείου, αυτή την φορά βάζοντας κάθε γραμμή στον πίνακα `strArray`.

Η αναπαράσταση της διαμοιραζόμενης μνήμης αποφάσισα ότι θα γίνει με ένα `struct` που θα περιέχει έναν πίνακα από 100 χαρακτήρες, καθώς και έναν ακέραιο αριθμό. Αυτή η αναπαράσταση διευκολύνει την υλοποίηση επικοινωνίας μητρικής διεργασίας και παιδιού καθώς το παιδί θα ζητάει έναν ακέραιο αριθμό και ο γονέας θα πρέπει να του επιστρέφει μία συμβολοσειρά.

Δημιουργώ την διαμοιραζόμενη μνήμη με την `shmget`, καθώς και τρία `set` από σημαφόρους, που έχουν ένα σημαφόρο το κάθε `set` με την `semget`. Επέλεξα αυτή την υλοποίηση για τους τρεις σημαφόρους αντί να κάνω ένα `set` με 3 σημαφόρους διότι κάνει πιο ευανάγνωστο τον κώδικα μου. Και για την δημιουργία της διαμοιραζόμενης μνήμης αλλά και για τους σημαφόρους δεν όρισα κλειδιά (όρισμα `IPC_PRIVATE`), καθώς δεν χρειάζεται αφού όλες οι ενέργειες γίνονται από μία μητρική διεργασία και τα παιδιά της και όχι δυο ξένες διεργασίες. Ορίζω αρχικές τιμές για τους 3 σημαφόρους, 0 για τον `sem_read`, 1 `sem_write`, 0 `sem_wait`. Ορίζω και το `struct sh` για την διαμοιραζόμενη μνήμη μέσω της `shmat`. Μετά κάνω ένα `for` κ επαναλήψεων όπου δημιουργώ τα κ παιδιά και εκτελώ

τον κώδικα αιτημάτων, όπου εκτελείται ένα for n επαναλήψεων. Μέσα στο for τον παιδιών ο σημαφόρος sem_read μειώνεται κατά 1. Αν είναι 0 τότε η διεργασία μπαίνει σε κατάσταση μπλοκ μέχρι να την ξεμπλοκάρει αργότερα ο πατέρας. Αν είναι διάφορος του 0 τότε προχωράει βάζοντας έναν τυχαίο αριθμό στον ακέραιο αριθμό της διαμοιραζόμενης μνήμης. Στη συνέχεια αυξάνει κατά ένα τον sem_write που τον εκμεταλεύεται αργότερα ο πατέρας, και μειώνει τον sem_wait κατά 1. Αν είναι 0 τότε η διεργασία μπαίνει σε κατάσταση μπλοκ μέχρι να την ξεμπλοκάρει αργότερα ο πατέρας. Αν είναι διάφορος του 0 τότε εκτυπώνει την συμβολοσειρά που έχει η διαμοιραζόμενη μνήμη. Αφού εκτελεστούν η n επαναλήψεις η διεργασία κάνει exit(). Στην αρχή και στο τέλος του for υπάρχουν δύο μεταβλητές χρόνο start, end οι οποίες μετράνε τον χρόνο εκτέλεσης των εντολών στο for. Αυτές χρησιμοποιούνται για τον υπολογισμό του μέσου χρόνο εκτέλεσης αυτών των εντολών για κάθε διεργασία.

Ο πατέρας κάνει μια for κ*n επαναλήψεων όπου σε αυτή κατεβάζει τον σημαφόρο sem_write κατά 1. Αν είναι 0 τότε η μητρική διεργασία μπαίνει σε κατάσταση μπλοκ μέχρι να την ξεμπλοκάρει αργότερα κάποιο από τα παιδιά. Αν είναι διάφορος του 0 τότε αντιγράφει την συμβολοσειρά του πίνακα strArray στο στοιχείο που είναι ο ακέραιος της διαμοιραζόμενης μνήμης, στη συμβολοσειρά της διαμοιραζόμενης μνήμης. Στη συνέχεια αυξάνει τους σημαφόρους sem_read, sem_wait κατά 1. Μετά το τέλος των επαναλήψεων ο πατέρας περιμένει να τελειώσουν όλα τα παιδιά με την wait(). Αφού γίνει αυτό αποδεσμεύονται οι σημαφόροι και η διαμοιραζόμενη μνήμη.

Ουσιαστικά η χρησιμότητα του σημαφόρου sem_write είναι να μπλοκάρει την μητρική διεργασία από το να αντιγράφει το στοιχείο strArray[sh → line] στην συμβολοσειρά της διαμοιραζόμενης μνήμης, μέχρι να εξασφαλιστεί ότι κάποιο από τα παιδιά έχει δώσει κάποια τιμή στην ακέραιο sh → line της διαμοιραζόμενης μνήμης.

Η χρησιμότητα του σημαφόρου sem_read είναι να μπλοκάρει όλα τα παιδιά εκτός από ένα έτσι ώστε να εξασφαλιστεί ότι ο

ακέραιος της διαμοιραζόμενης μνήμης θα πάρει μία μία τιμή από όλες της διεργασίες.

Η χρησιμότητα του σημαφόρου `sem_wait` είναι να μπλοκάρει τα παιδιά από το να εκτυπώσουν την συμβολοσειρά της διαμοιραζόμενης μνήμης, μέχρι να εξασφαλιστεί ότι ο πατέρας έχει δώσει κάποια τιμή σε αυτή.

Παράδειγμα:

Όπως τρέχει το πρόγραμμα, ο πατέρας πάει να κάνει `down()` τον σημαφόρο `sem_write` και μπλοκάρει αφού ο σημαφόρος είναι 0. Το πρώτο παιδί κάνει `down()` τον σημαφόρο `sem_read` και πετυχαίνει αφού η τιμή του είναι 1. Όλα τα υπόλοιπα παιδιά μπλοκάρουν όταν πάνε να εκτελέσουν την ίδια εντολή. Το πρώτο παιδί δίνει μία τυχαία τιμή στον ακέραιο `sh → line` της διαμοιραζόμενης μνήμης και μετά κάνει `up()` τον σημαφόρο `sem_write` ξεμπλοκάροντας τον πατέρα. Μετά κάνει `down()` τον σημαφόρο `sem_wait` και μπλοκάρει αφού είναι 0. Ο πατέρας αντιγράφει την τιμή του στοιχείου `strArray[sh → line]` στη συμβολοσειρά της διαμοιραζόμενης μνήμης και κάνει `up()` τον σημαφόρο `sem_read`, ξεμπλοκάροντας κάποιο από τα παιδιά, και κάνει `up()` τον σημαφόρο `sem_wait` ξεμπλοκάροντας το πρώτο παιδί. Το πρώτο παιδί τώρα εκτυπώνει τον συμβολοσειρά της διαμοιραζόμενης μνήμης. Το πρόγραμμα συνεχίζει....