Λειτουργικά συστήματα K22

ΤΕΚΜΗΡΙΩΣΗ 3ΗΣ ΕΡΓΑΣΙΑΣ

«Υλοποίηση κοινής μνήμης και σηματοφόρων στο λειτουργικό σύστημα xv6»

Ονοματεπώνυμο: Κώστας Χατζόπουλος **ΑΜ:** 1115201300202

ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΔΟΜΩΝ ΠΟΥ ΧΡΕΙΑΣΤΗΚΑΝ

Για την υλοποίηση κοινής μνήμης στο λειτουργικό σύστημα xv6 είναι απαραίτητο να αποθηκεύονται σε έναν πίνακα μεγέθους 32 οι σελίδες (pages[NSHMPG]), συγκεκριμένα κάθε στοιχείο αυτού του πίνακα είναι τύπου shmpg, δηλαδή η δομή οπου περιέχει μια σελίδα καθώς και πληροφορίες για αυτή.

Ποιο αναλυτικά η δομή της σελίδας περιέχει το κλειδί αυτής (sh_key_t key) τη φυσική μνήμη όπου βρίσκεται (char *pa) και από πόσες διεργασίες χρησιμοποιείται (int usages). Για την εξασφάλιση της ασφαλής προσπέλασης αυτού του πίνακα, τον τοποθέτησα μέσα σε μια νέα δομή με όνομα shmtable όπου συνοδεύεται μαζί με μια κλειδαριά (struct spinlock lock) και κάθε φορά που υπάρχει η ανάγκη προσπέλασής του, γίνεται **acquire** και αντίστοιχα **release** αυτής της κλειδαριάς.

Για το κλειδί κάθε σελίδας αποθηκεύω σε μια δομή το μέγεθός του καθώς και το ίδιο το κλειδί σε ένα πίνακα από bytes-characters μεγέθους SHMKEYSIZE.

Για τη δομή του σηματοφόρου υπάρχει ένας μετρητής καθώς και μια κλειδαριά.

ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ:

Την ώρα που ξεκινάει ο πυρήνας, γίνεται αρχικοποίηση των δομών της κοινής μνήμης κατάλληλα αμέσως πριν ξεκινήσει ο scheduler με τη βοήθεια της συνάρτησης shminit, αμέσως μετά εμφανίζεται μήνυμα ολοκλήρωσης (με πράσινα γράμματα).

Όταν ο χρήστης ζητήσει μια κοινόχρηστη σελίδα, πρέπει πρώτα να φτιάξει ένα κλειδί, να το κάνει αρχικοποίηση με τη βοήθεια του system call shm_key_init ώστε να το δώσει στην shmget και τέλος να πάρει τη σελίδα που ζητάει.

Η shmget πρώτα εξετάζει τον shmtable για να δει αν υπάρχει σελίδα με το συγκεκριμένο κλειδί, αν υπάρχει τότε ενημερώνεται κατάλληλα η μεταβλητή page ώστε να κληθεί αργότερα η shmpgmap με όρισμα τη μεταβλητή αυτή και να γίνει το mapping της φυσικής διεύθυνσης στην κατάλληλη θέση του εικονικού address space της διεργασίας. Αν δεν υπάρχει τότε δημιουργεί μια καινούργια σελίδα σε κάποια από τις κενές θέσεις του πίνακα shmtable μέσω της pgalloc και πάλι ενημερώνει την μεταβλητή page.

Σε κάθε περίπτωση καλείται η shmpgmap όπου θα αναλάβει να βρει την πρώτη ελεύθερη «τρύπα» στον εικονικό χώρο της τρέχουσας διεργασίας και να κάνει το mapping της φυσικής σελίδας στην αντίστοιχη εικονική που βρήκε. (Σε περίπτωση πληρότητας επιστρέφεται 0). Τέλος, με αυτό τον τρόπο επιστρέφεται στο χρήστη η εικονική διεύθυνση όπου πλέον «καθρεφτίζει» στη φυσική διεύθυνση της αναφερόμενης σελίδας.

Η shmrem αναλαμβάνει να αφαιρέσει τη σελίδα με το συγκεκριμένο κλειδί από το address space της διεργασίας που την κάλεσε, τέλος εξετάζει τη σελίδα αυτή ώστε να διαπιστώσει αν τη συγκεκριμένη χρονική στιγμή είναι η τελευταία που έκανε χρήση αυτής της σελίδας. Αν είναι τότε η σελίδα διαγράφεται από τη φυσική μνήμη και αρχικοποιείται ξανά το αντίστοιχο entry που τη φιλοξενούσε στον πίνακα shmtable.

ΠΕΡΙΓΡΑΦΗ ΤΩΝ WORKLOADS:

Το workload που χρησιμοποιώ ζητάει μια κοινόχρηστη σελίδα, κατασκευάζει ένα σηματοφόρο τον αποθηκεύει στην αρχή της κοινή μνήμης. Αμέσως μετά αρχικοποιεί τη μεταβλητή pool και την αποθηκεύει μέσα στην κοινή μνήμη ακριβώς δίπλα από το σηματοφόρο. Τέλος δημιουργεί έναν αριθμό από **child processes** τα οποία ζητάνε από τη σελίδα τη τιμή της μεταβλητής και του σηματοφόρου, εκτελούν ταυτόχρονα μια αφαίρεση από το pool και το αποθηκεύουν εκ νέου στην κοινή μνήμη. Όλο αυτό για να γίνει σωστά χρειάζεται οπωσδήποτε την υποστήριξη συγχρονισμού ώστε να αποφευχθεί η ταυτόχρονη προσπέλαση (read-write) δεδομένων καθώς και η καταστροφή αυτών με αποτέλεσμα τον λάθος υπολογισμό.

Δηλαδή κάθε χρονική στιγμή που εκτελείται κώδικας μέσα στην κρίσιμη περιοχή **critical section**, πρέπει μόνο μια διεργασία τη φορά να εκτελείται για την αποφυγή λαθών ή καταστροφής δεδομένων κτλ.