Λειτουργικά Συστήματα/Περίοδος 2021-2022
2η Εργασία README

Παναγιώτα Γύφτου, ΑΜ.: 1115201900318

Ιανουάριος 2022

↬ Τροποποίηση ***uvmcopy()***

Αρχικά για κάθε σελίδα βρίσκω το pte (page table entry) της μέσω της *walk()* , έπειτα γίνεται έλεγχος αν η pte είναι έγκυρη (*pte & PTE_V).Συνεχίζουμε στο σημείο όπου τροποποιώ την ***uvmcopy()*.** Εδώ επεξεργάζομαι τα κομμάτια των σημαιών της εικονικής σελίδας PTE, συγκεκριμένα αρχικά ορίζω να μη γίνεται χρήση της σημαίας εγγραφής PTE_W (*flags &= (~PTE_W)*),βάζω δηλαδή το flag ίσο με 0.Η τελευταία επεξεργασία ως προς τις σημαίες είναι να θέσω σε χρήση τη σημαία αντιγραφής κατά την εγγραφή PTE_COW (*flags |=PTE_COW*) .Έτσι με αυτόν τον τρόπο πετυχαίνω την σελίδα που αντιστοιχεί στο pagetable του παιδιού να είναι read only, και ορίζοντας το flag να είναι read only.Στην συνέχεια ανανεώνω και το παλιό pte με το καινούριο (**pte=PA2PTE(pa) |flags*).Μετά σχολιάζω το κομμάτι τις kalloc όπου εκχωρεί φυσικές σελίδες, ώστε να αποφευχθεί η αντιγραφή της μνήμης στο παιδί.Έπειτα τροποποιώ κατάλληλα το mappage ώστε να μη δηλώνεται η καινούρια σελίδα mem (η οποία και δεν δημιουργήθηκε, γιατί δεν καλέσαμε την kalloc()),αλλά η υπάρχουσα (pa).Τέλος καλώντας την *increase_reference_cnt()*, αυξάνεται ο αριθμός αναφοράς για την συγκεκριμένη σελίδα (η αρχική διευθυνση pa δε τροποποιείται).

Η *increase_reference_cnt()* βρίσκεται στο αρχείο cow.c, η λειτουργία της είναι να αυξάνει το reference counter, τον μετρητή αναφορών.Ο μετρητής αναφοράς αυξάνεται, γιατί πλέον ο γονέας και το παιδί δείχνουν στο ίδιο, οπότε αυξάνονται οι αναφορές.

↬ Τροποποίηση ***usertrap()***

Εδώ πρέπει να γίνεται έλεγχος και στην περίπτωση που γίνονται σφάλματα σε μια σελίδα CoW.Για την τροποποίηση της ***usertrap()*** προσθέτω μια επιπλέον συνθήκη για το αν το r_scause() έχει τιμή 15, (επειδή ο μηχανισμός COW εξετάζεται μόνο όταν ενεργοποιείται r_scause()=15) .Αρχικά στο σώμα της συνθήκης καλώ την *cow_f() ,* που έχω δημιουργήσει στο αρχείο vm.c (vm.c: στο αρχείο αυτό βρίσκεται το μεγαλύτερος μέρος κώδικα σε ότι αναφορά το χειρισμό χώρων διευθύνσεων και πινάκων σελίδων), και της δίνω τα ορίσματα pagetable και την εικονική διεύθυνση, που δημιούργησε αυτό το page fault (***cow_f(p->pagetable, r_stval())***), όπου θα ελέγξει

αν η σελίδα είναι read only, γιατί το trap έχει προκληθεί μόνο από read only σελίδες.Στην συνέχεια η *cow_f* αντιγράφει την read only σελίδα σε μια read write σελίδα.Η read only που πλέον δεν την χρησιμοποιεί η συγκεκριμένη διεργασία θα πρέπει να αφαιρεθεί.

Η συνάρτηση *cow_f ()* αρχικά το πρώτο που ελέγχει είναι αν η virtual address είναι έγκυρη , αυτός ο unsigned αριθμός πρέπει να είναι μικρότερος από την MAXVA (βιβλίο σελ.26) αν είναι μεγαλύτερος τότε τερματίζεται (***if (va >= MAXVA){return 0;}***) αλλιώς συνεχίζει.Στην συνέχεια μέσω του pagetable, της virtual address και με την βοήθεια της *walk()* θα βρούμε το page table entry που αντιστοιχεί στην συγκεκριμένη σελίδα (***pte=walk(pagetable, va, 0)***) και μετά θα ελέγχει αν αυτό το pte είναι έγκυρο (****pte & PTE_V***), δηλαδή αν έχει γίνει allocate από το σύστημα.Έπειτα πρέπει να γίνει και ο έλεγχος για το εάν είναι user level (user level: είναι τα κομμάτια μνήμης που έχουν ανατεθεί για την μνήμη), τα cow fault πρέπει να είναι μόνο τύπου user (****pte & PTE_U***).Συνεχίζουμε ελέγχοντας αν είναι read only, ελέγχοντας σε πρώτη φάση αν το flag του PTE_W είναι 0 (***if ((*pte & PTE_W) == 0)***), δηλαδή δεν είναι σελίδα εγγραφής και σε δεύτερη φάση μέσα στο σώμα της γίνεται έλεγχος για το εάν είναι η σημαία PTE_COW διάφορη του 0, ώστε να είναι μια σελίδα copy on write, αν δεν είναι γίνεται τερματισμός.Μετά πρέπει να δημιουργήσουμε μια καινούρια σελίδα, να καλέσουμε δηλαδή την *kalloc(),* να αντιγράψουμε τα περιεχόμενα της υφιστάμενης σελίδας στην καινούρια.Αν η *kalloc()* αποτύχει , επιστρέφουμε τιμή 0 που υποδεικνύει αποτυχία και τερματίζουμε.Στην συνέχεια με την memmove γίνεται η ανάθεση ενός καινούριου physical address και αντιγράφει η memmove το physical address του πρώτου στο δεύτερο (physical address), ουσιαστικά αντιγράφεται όλη η σελίδα.Με την uvmunmap γίνεται η απελευθέρωση της φυσικής μνήμης ,που πλέον δεν χρειάζεται ώστε να μη δεσμεύουμε χώρο χωρίς να χρησιμοποιείται.Έπειτα με την mappage δηλώνεται η καινούρια σελίδα mem αν η mappage αποτύχει με την kfree ελευθερώνουμε την mem και επιστρέφουμε τιμή 0 για να ενημερώσουμε.Στο τέλος επιστρέφουμε την φυσική διεύθυνση.

↬ Τροποποίηση ***kalloc()***

Η *kalloc()* εκχωρεί φυσικές σελίδες.Έχει μια λίστα σελίδων φυσικής μνήμης που είναι διαθέσιμες για καταχώρηση.Η τροποποίηση, η οποία γίνεται είναι όταν ανατίθεται αρχικά μία σελίδα ο μετρητής αναφορών να αρχικοποιείται με 1, αυτό επιτυγχάνεται μέσω της (*increase_reference_cnt()),* ελέγχοντας πρώτα αν είναι 0 μέσω της (*reference_cnt()) .*

Η *reference_cnt() ,* βρίσκεται στο αρχείο cow.c και αυτό που κάνει είναι να επιστρέφει το πλήθος αναφορών της σελίδας.

↬ Τροποποίηση ***kfree()***

Η *kfree()* πριν την τροποποίηση της αυτό που κάνει είναι να αφαιρεί την σελίδα από τις διαθέσιμες και να την προσθέτει στις σελίδες με τις free. Την τροποποιούμε έτσι ώστε πρώτα να αφαιρεί (μείον) ένα από το reference count και όταν το reference count γίνει μηδέν τότε να προχωράει στην δουλειά που αναφέρθηκε προηγουμένως (δηλ. στην δουλειά που έκανε η kfree() πριν την τροποποίηση της).Υπάρχει περίπτωση να μην έχει γίνει μόνο μια αναφορά στην συγκεκριμένη σελίδα γι' αυτό γίνεται αναδρομικά η αφαίρεση , δηλαδή η κλήση της *decrease_reference_cnt()* μέχρι ο reference counter να είναι ίσος με το 0.

Η *decrease_reference_cnt(),* βρίσκεται στο αρχείο cow.c, η λειτουργία της είναι να μειώνει το reference counter, τον μετρητή αναφορών και να επιστρέψει το καινούριο πλήθος αναφορών.Ο μετρητής αναφοράς μειώνεται, οι αναφορές σε μια σελίδα μειώνονται κατά 1.

↬ Τροποποίηση ***freerange()***

Δουλειά της ***freerange()*** είναι να αρχικοποιεί την αρχική ανάθεση μνήμης στο σύστημα με την βοήθεια της *kfree()*.Η τροποποίηση που γίνεται είναι ότι αφού έχει τροποποιηθεί η *kfree(),* θα πρέπει για κάθε σελίδα να αρχικοποιεί το reference counter ίσο με το 1(***increase_reference_cnt((uint64)p)***) και μετά να γίνεται η *kfree().*Η *kfree()* έχει δύο λειτουργίες: αποδέσμευση μνήμης και αρχικοποίησης αρχικής μνήμης.

↬ Τροποποίηση ***copyout()***

Η *copyout()* αντιγράφει σελίδες από το kernel στο user memory.Η *walkaddr()* παίρνει απευθείας το physical address από ένα virtual address, χωρίς να περάσει από το MMU του hardware του risc-V (μονάδα που είναι ενσωματωμένη στον επεξεργαστή, που ο επεξεργαστής μέσω αυτής διαχειρίζεται την μετάφραση, δηλαδή θα κάνει την μετάφραση του virtual address στο physical address), το οποίο σημαίνει ότι δεν έχει προκληθεί σφάλμα σελίδας.Η *walkaddr()* κάνει την δουλειά που κάνει η MMU , δηλαδή χρησιμοποιεί τα pagetables , όμως σε επίπεδο software.Τροποποιούμε την *copyout()* κατάλληλα ώστε να δέχεται και CoW σελίδες, κάνοντας την ίδια λειτουργία.Αυτό που αλλάζουμε είναι την ***walkaddr()***, την αντικαθιστούμε με την ***cow_f().***Όπου η ***cow_f()*** έχει την ίδια λειτουργία με την walkaddr() , με την διαφορά ότι επεξεργάζεται και τις CoW σελίδες .