

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Αθηνών
Λειτουργικά Συστήματα (Κ22) / Περίοδος 2021-2022
2^η Εργασία

**Υποστήριξη Αντιγραφής κατά την Εγγραφή (Copy On Write – COW)
στο xν6**

Το πρόβλημα - Η υπάρχουσα υλοποίηση της κλήσης συστήματος `fork()` στο xν6 αντιγράφει όλη τη μνήμη (user-space) της γονικής διεργασίας στο παιδί. Επιπλέον, η εργασία αυτή συχνά είναι άχρηστη, ειδικά όταν η `fork()` ακολουθείται από μία `exec()` στο παιδί, με την οποία απορρίπτεται η αντιγραμμένη μνήμη, χωρίς καν να έχει χρησιμοποιηθεί στο μεγαλύτερο τμήμα της. Όμως αν και οι δύο διεργασίες (γονική, παιδί) χρησιμοποιούν μία σελίδα και η μία ή και οι δύο από αυτές γράψουν σε αυτή, τότε απαιτείται ένα ξεχωριστό αντίγραφο για κάθε μία από αυτές

Η λύση - Ο στόχος της υποστήριξης αντιγραφής κατά την εγγραφή (copy on write – cow) κατά τη `fork()` είναι να αναβάλλει την ανάθεση και αντιγραφή φυσικών σελίδων μνήμης για το παιδί μέχρι πραγματικά να χρειαστεί να τις χρησιμοποιήσει αν όντως τις χρειαστεί.

Η `fork()` με υποστήριξη αντιγραφής κατά την εγγραφή δημιουργεί μόνο ένα πίνακα σελίδων για το παιδί με τις εγγραφές του πίνακα σελίδων (Page Table Entries – PTE) να δείχνουν στις φυσικές σελίδες του γονέα. Οι (user space) PTEs δηλώνονται ως μη εγγράψιμες τόσο στον γονέα όσο και στο παιδί. Όταν οποιαδήποτε από αυτές τις διεργασίες προσπαθήσει να γράψει σε μία από αυτές τις CoW σελίδες, η CPU θα προκαλέσει ένα σφάλμα σελίδας. Ο χειριστής σφαλμάτων σελίδων στον πυρήνα θα ανιχνεύσει αυτή την περίπτωση, θα αναθέσει μία σελίδα φυσικής μνήμης για τη διεργασία που προκάλεσε το σφάλμα, θα αντιγράψει την αρχική σελίδα στη νέα και θα τροποποιήσει τη σχετική PTE στη διεργασία που προκάλεσε το σφάλμα, αυτή τη φορά με την PTE ως εγγράψιμη. Όταν επιστρέψει ο χειριστής σφαλμάτων σελίδας του πυρήνα, η διεργασία χρήστη θα μπορεί να τροποποιήσει (γράψει) το δικό της αντίγραφο της σελίδας.

Η υποστήριξη CoW στη `fork()` κάνει την απελευθέρωση φυσικών σελίδων που έχουν ανατεθεί σε user space μνήμη πολύπλοκότερη. Μία φυσική σελίδα μπορεί να αναφέρεται σε πολλαπλούς πίνακες σελίδων διεργασιών και θα πρέπει να απελευθερώνεται μόνο όταν καταργείται και η τελευταία αναφορά σε αυτή.

Υλοποίηση αντιγραφής κατά την εγγραφή (copy on write)

Η εργασία αφορά την υλοποίηση αντιγραφής κατά την εγγραφή (copy on write) στον πυρήνα xν6. Για δοκιμή ελέγχου της υλοποίησής σας, παρέχονται τα προγράμματα ελέγχου `cowtest` και `usertests`. Το `cowtest` εκτελεί διάφορες δοκιμές, εκ των οποίων όλες αποτυγχάνουν στην αρχική μορφή του πυρήνα. Αρχικά θα δείτε:

```
$ cowtest
simple: fork() failed
$
```

Το “simple” test αναθέτει σε μία διεργασία περισσότερο από το μισό της διαθέσιμης φυσικής μνήμης και στη συνέχεια καλεί τη `fork()`. Η `fork()` αποτυγχάνει επειδή δεν υπάρχει αρκετή φυσική μνήμη για δώσει στο παιδί ένα πλήρες αντίγραφο της μνήμης της γονικής διεργασίας.

Όταν ολοκληρώσετε την εργασία, ο πυρήνας θα πρέπει να περνάει όλα τα τεστ στα προγράμματα `cowtest` και `usertests`, δηλαδή:

```
$ cowtest
simple: ok
simple: ok
three: zombie!
ok
three: zombie!
ok
three: zombie!
ok
file: ok
ALL COW TESTS PASSED
$ usertests
...
ALL TESTS PASSED
$
```

Υποδείξεις για την υλοποίηση της εργασίας:

1. Τροποποιήστε την `uvmcopy()` ώστε να αντιστοιχίζει τις φυσικές σελίδες μνήμης του γονέα στο παιδί, αντί να αναθέτει νέες σελίδες. Μηδενίστε το flag `PTE_W` στις PTE και του γονέα και του παιδιού.
2. Τροποποιήστε την `usertrap()` για να αναγνωρίζει σφάλματα σελίδας. Όταν συμβαίνει ένα σφάλμα σελίδας σε μία σελίδα CoW, να αναθέτει μία νέα σελίδα μέσω της `kalloc()`, να αντιγράφει την παλιά σελίδα στη νέα και να εγκαθιστά τη νέα σελίδα στην PTE με το `PTE_W` bit ίσο με 1.
3. Βεβαιωθείτε ότι οι φυσικές σελίδες απελευθερώνονται μόνο όταν μηδενίζονται οι αναφορές σε αυτή από PTE. Θα μπορούσατε να κρατάτε ένα μετρητή αναφορών για κάθε φυσική σελίδα, όπου θα μετρούνται οι πίνακες σελίδων που αναφέρονται σε αυτή τη σελίδα. Όταν ανατίθεται αρχικά μία σελίδα μέσω της `kalloc()`, ο μετρητής αναφορών αρχικοποιείται στην τιμή 1. Αυξάνετε τον μετρητή όταν η `fork()` δημιουργεί ένα παιδί που μοιράζεται τη σελίδα και μειώνετε τον μετρητή όταν μία διεργασία αφαιρεί τη σελίδα από τον πίνακα σελίδων της. Η `kfree()` θα πρέπει να επιστρέφει τη σελίδα στη λίστα με τις διαθέσιμες σελίδες μόνο όταν ο μετρητής μειωθεί στο 0. Μπορείτε να χρησιμοποιήσετε έναν πίνακα ακεραίων για την αποθήκευση των μετρητών και μεγέθους ίσο με τον συνολικό αριθμό των φυσικών σελίδων στο σύστημα.
4. Τροποποιήστε την `copyout()` ώστε να χρησιμοποιεί τον ίδιο μηχανισμό όπως τα σφάλματα σελίδας όταν συναντά μία σελίδα CoW.

Υποδείξεις:

- Μπορεί να είναι χρήσιμη η καταγραφή για κάθε PTE αν είναι αντιστοίχιση CoW. Μπορείτε να χρησιμοποιήσετε τα bits RSW (reserved for software) στις PTE του RISC-V για αυτό τον σκοπό.
- Θα χρειαστείτε τις πληροφορίες στο `kernel/riscv.h`
- Αν συμβεί σφάλμα σελίδας CoW και δεν υπάρχει διαθέσιμη μνήμη, η διεργασία θα πρέπει να τερματιστεί.

Κώδικας

Χρησιμοποιήστε τον κώδικα για την εργασία ως εξής:

```
$ git clone git://gallagher.di.uoa.gr/xv6-project-2021
Cloning into 'xv6-project-2021'...
...
$ cd xv6-project-2021
```

Το αποθετήριο (repository) `xv6-project-2021` προσθέτει λίγη επιπλέον λειτουργικότητα σε σχέση με το κύριο. Για να δείτε τις αλλαγές που έχουν γίνει μπορείτε να δώσετε την εντολή

```
$ git log
```

Τα αρχεία που θα χρειαστείτε για αυτή την εργασία διανέμονται μέσω του συστήματος ελέγχου πηγών κώδικα git. Μπορείτε να βρείτε πληροφορίες για το git στο [βιβλίο git](#) ή σε άλλες δημόσιες πηγές. Το git επιτρέπει να διατηρείτε πληροφορία για όλες τις αλλαγές που έχετε κάνει στον κώδικα. Για παράδειγμα, αν τελειώσετε ένα μέρος της εργασίας και θέλετε να καταχωρήσετε τοπικά τις αλλαγές σας, μπορείτε να καταγράψετε (commit) τις αλλαγές σας μέσω της εντολής

```
$ git commit -am 'my solution for cow implementation project'
Created commit 60d2135: my solution for util lab exercise 1
1 files changed, 1 insertions(+), 0 deletions(-)
$
```

Ημερομηνία Παράδοσης: 23/Ιαν/2021

Τρόπος παράδοσης: υποβολή στο eclass, θα πρέπει να παραδοθεί ένα αρχείο tar με περιεχόμενο όλα τα σχετικά αρχεία.

Συνοδευτικό υλικό: τεκμηρίωση 3-4 σελίδων που να εξηγεί τον τρόπο με τον οποίο εργαστήκατε.

Υλοποίηση: η εργασία είναι ατομική.

Η εργασία θα εξεταστεί στα συστήματα του Τμήματος σύμφωνα με πρόγραμμα που θα ανακοινωθεί μετά την ημερομηνία παράδοσης.