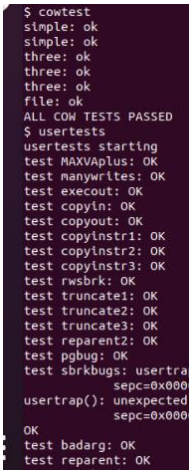
ΟΝΟΜΑ: Μαρία Μήλιου

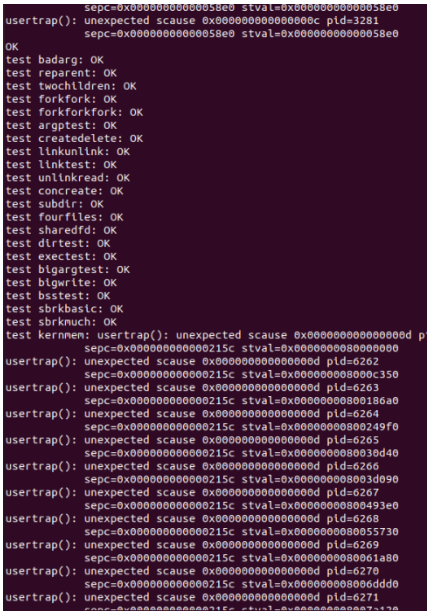
Α.Μ.: 1115201300101

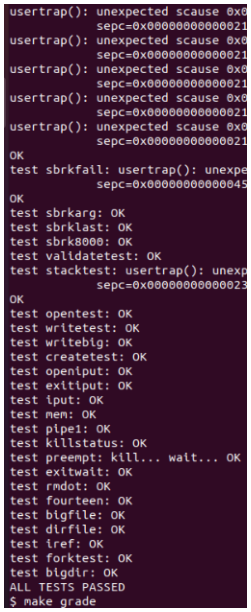
Λειτουργικά Συστήματα Εργασία 2η

Εκτέλεση

Έτρεξα τον κώδικα σε vm Ubuntu 20.04. Μετά τις αλλαγές ο πυρήνας περνάει επιτυχώς όλα τα τεστ στα δύο προγράμματα cowtest και usertests. Παρακάτω παρουσιάζονται τα screenshots από τα αποτελέσματα των τεστ.

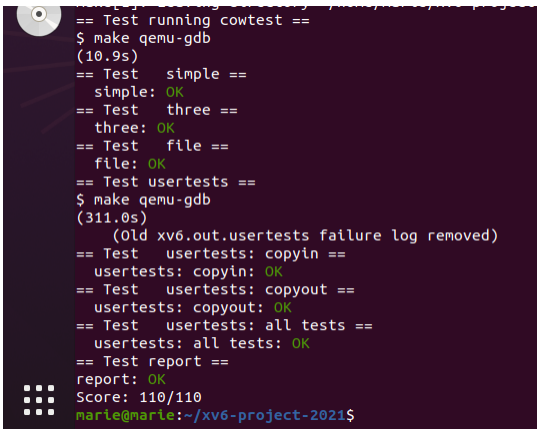






1 2 3

Τρέχοντας το make grade όμως κάνει timeout στα 300 s και εμφανίζει ότι δεν περνάει το usertests. Τροποποιώντας τον χρόνο του timeout στο grade-lab-cow line 32 σε 400s τερματίζει δίνοντας όλους τους πόντους. Στο αρχείο όμως που θα στείλω θα αφήσω το timeout στα 300s. Παρακάτω το screenshot (400s).



Αλλαγές

Παρακάτω επεξηγούνται οι αλλαγές που έχω κάνει στα αρχεία που χρειάστηκε

- ***defs.h***
  - Δήλωση του πίνακα με τους μετρητές αναφορών και του reflock έτσι ώστε τα αρχεία που θέλω να έχουν πρόσβαση σε αυτά συμπεριλαμβάνοντας την βιβλιοθήκη αυτή(lines 11-12). Έχουν δηλωθεί σαν extern γιατί ο ορισμός και η αρχικοποίηση θα γίνει στο *kalloc.c*.
  - Δήλωση της συνάρτησης void refinit(void) η οποία αρχικοποιεί το reflock.(line 68)
  - Δήλωση της walk από vm.c για να μπορεί να χρησιμοποιηθεί στο trap.c. (line 176)

- ***kalloc.c***
  - Ορισμός του πίνακα με τους μετρητές αναφορών με μέγεθος το άνω φράγμα της φυσικής μνήμης προς το μέγεθος της σελίδας (PHYSTOP/PGSIZE). (line 34)
  - Ορισμός της κλειδαριάς *reflock* σαν *struct spinlock. (line 35)*
  - Ορισμός της refinit() η οποία αρχικοποιεί την κλειδαριά με όνομα refcount. (lines 38-42)
  - *freerange/lines 51-52*:
    Αρχικοποίηση των μετρητών αναφοράς για κάθε σελίδα σε ένα. Αυτό γιατί κατά την εκκίνηση ο πυρήνας απελευθερώνει όλες τις σελίδες καλώντας την kfree πριν την αρχικοποίησή του στο kalloc().
  - *kfree*
    Μια σελίδα θα πρέπει να απελευθερώνεται από την kfree μόνο όταν ο μετρητής αναφοράς της είναι 0. Οπότε στον κώδικά της αρχικά μειώνει το refcounter της σελίδας και στη συνέχεια ελέγχει αν έχει μηδενιστεί. Αν ναι τότε απελευθερώνει την σελίδα με το τρόπο που έκανε και αρχικά, διαφορετικά δεν κάνει κάτι παραπάνω.
  - *kalloc*
    Η μόνη διαφορά είναι ότι αρχικοποιεί τους μετρητές αναφοράς σε 1 όταν αναθέτει μια σελίδα. (lines 111-113)

- ***main.c***
  Καλεί την refinit() στη main για να αρχικοποιήσει το reflock όπως κάνει και με τα υπόλοιπα lock. (line 32)

- ***trap.c***
  - *usertrap*
    Προσθέτω και την περίπτωση όπου η usertrap αναγνωρίζει σφάλματα σελίδας από CoW. Αν ο καταχωρητής scause είναι ίσος με 15 εκτελεί τα παρακάτω βήματα:
    - Ελέγχει αν το virtual address (va) δεν είναι μεγαλύτερο ή ίσο από το επιτρεπτό (MAXVA), διαφορετικά κάνει exit με -1.
    - Βρίσκει το page table entry (pte) της virtual address που προκάλεσε το page fault καλώντας την walk με ορίσματα το page table της διεργασίας, και το va. Η ενέργεια που προκάλεσε το page fault βρίσκεται στον καταχωρητή stval και κάνωντας PGROUNDDOWN βρίσκουμε και ποια σελίδα από την εικονική μνήμη το προκάλασε (va). Αν αποτυγχάνει η walk προκαλούμε panic.
    - Αν το PTE_W είναι ίσο με 0 και PTE_R ,PTE_U,PTE_V ίσο με 1, αναθέτει μια νέα σελίδα μέσω της kalloc (), αντιγράφει την παλιά σελίδα στην νέα, θέτει το PTE_W σε 1, διαγράφει την αντιστοίχιση που είχε με την παλιά σελίδα και αντιστοιχίζει με την νέα σελίδα, και μειώνει τον μετρητή αναφοράς της παλιάς σελίδας κατά 1. Αν αποτύχει η kalloc σκοτώνει την διεργασία και κάνει exit με -1. Αν ο μετρητής αναφοράς της παλιάς σελίδας είναι ίσος με 0 απελευθερώνει την σελίδα καλώντας την kfree.

- ***vm.c***
  - *uvmcopy*
    Για κάθε φυσική σελίδα μνήμης του γονέα:
    - Μηδενίζει το flag PTE_W του page table entry που αντιστοιχίζεται με την εκάστοτε φυσική σελίδα του γονέα.
    - Αντιστοιχίζει το παιδί με την σελίδα αυτή καλώντας την mappages με ορίσματα μεταξύ άλλων το page table του παιδιού και flags ίδια με γονέα όπου PTE_W = 0.
    - Αυξάνει το μετρητή αναφοράς στη σελίδας αυτή κατά 1.
  - *copyout*
    Κατά την αντιγραφεί στο destination virtual address:
    - Ελέγχει αν το va είναι στα πλαίσια του επιτρεπτού.
    - Βρίσκει το pte και τσεκάρει το PTE_W είναι ίσο με 0 και PTE_R ,PTE_U,PTE_V ίσο με 1.
    - Αν περάσει τον έλεγχο αναθέτει μια καινούρια σελίδα μέσω της kalloc, αντιγράφει την παλιά σελίδα στην νέα, θέτει το PTE_W σε 1, διαγράφει την αντιστοίχιση που είχε με την παλιά σελίδα και αντιστοιχίζει με την νέα σελίδα, και μειώνει τον μετρητή αναφοράς της παλιάς σελίδας κατά 1. Αν αποτύχει η kalloc επιστρέφει

-1. Αν ο μετρητής αναφοράς της παλιάς σελίδας είναι ίσος με 0 απελευθερώνει την σελίδα καλώντας την kfree.

Σημείωση:

Όπως ειπώθηκε και στο φροντιστήριο δεν χρησιμοποίησα τα επιπλέον bits που προσφέρονται για να προσδιορίσω ότι μια σελίδα είναι CoW αλλά χρησιμοποιώ το PTE_W flag.