

## Λειτουργικά Συστήματα

### Σχολιασμός Εργασίας 2

**Όνομα:** Ιωάννης Σαχλάνης

**A.M.:** 1115 2018 00292

### Σχολιασμός εργασίας

Η άσκηση υλοποιήθηκε ακολουθώντας τις οδηγίες που περιγράφονται στην εκφώνηση. Πιο συγκεκριμένα, ξεκίνησα με την υλοποίηση της vmprint συνάρτησης στο αρχείο kernel/vm.c, με την αντίστοιχη προσθήκη της στο defs.h. Για την ευκολότερη κλήση της παραπάνω συνάρτησης, υλοποίησα και την void printwalk(pagetable\_t pagetable, int depth).

Με βάση την παραπάνω συνάρτηση είδα να εκτυπώνεται ο πίνακας σελίδων για την πρώτη διεργασία χρήστη (init):

```
page table 0x0000000087f6c000
..0: pte 0x0000000021fda001 pa 0x0000000087f68000
.. ..0: pte 0x0000000021fd9c01 pa 0x0000000087f67000
.. .. ..0: pte 0x0000000021fda41b pa 0x0000000087f69000
.. .. ..1: pte 0x0000000021fd9817 pa 0x0000000087f66000
.. .. ..2: pte 0x0000000021fd9407 pa 0x0000000087f65000
.. .. ..3: pte 0x0000000021fd9017 pa 0x0000000087f64000
..255: pte 0x0000000021fdac01 pa 0x0000000087f6b000
.. ..511: pte 0x0000000021fda801 pa 0x0000000087f6a000
.. .. ..510: pte 0x0000000021fdd007 pa 0x0000000087f74000
.. .. ..511: pte 0x0000000020001c0b pa 0x0000000080007000
init: starting sh
page table 0x0000000087f5f000
..0: pte 0x0000000021fd6c01 pa 0x0000000087f5b000
.. ..0: pte 0x0000000021fd6801 pa 0x0000000087f5a000
.. .. ..0: pte 0x0000000021fd701b pa 0x0000000087f5c000
.. .. ..1: pte 0x0000000021fd641b pa 0x0000000087f59000
.. .. ..2: pte 0x0000000021fd6017 pa 0x0000000087f58000
.. .. ..3: pte 0x0000000021fd5c07 pa 0x0000000087f57000
.. .. ..4: pte 0x0000000021fd5817 pa 0x0000000087f56000
..255: pte 0x0000000021fd7801 pa 0x0000000087f5e000
.. ..511: pte 0x0000000021fd7401 pa 0x0000000087f5d000
.. .. ..510: pte 0x0000000021fdb407 pa 0x0000000087f6d000
.. .. ..511: pte 0x0000000020001c0b pa 0x0000000080007000
```

Στη συνέχεια έκανα τροποποιήσεις στα εξής αρχεία:

- kernel/defs.h: προσθήκη πρωτοτύπων νέων συναρτήσεων

- kernel/sysproc.c: Μεταβολή της sys\_sbrk χωρίς να εκχωρείται μνήμη
- kernel/trap.c: Γίνεται σωστός χειρισμός των Pagefaults
- kernel/vm.c: Υλοποίηση της lazyalloc συνάρτησης με τις αντίστοιχες αλλαγές όπου αλλού χρειαζόταν.

### Ενδεικτική εκτέλεση

```
$ make qemu
...
$ lazytests
page table 0x0000000087f51000
..0: pte 0x0000000021fd3401 pa 0x0000000087f4d000
.. ..0: pte 0x0000000021fd3001 pa 0x0000000087f4c000
.. .. ..0: pte 0x0000000021fd381b pa 0x0000000087f4e000
.. .. ..1: pte 0x0000000021fd2c17 pa 0x0000000087f4b000
.. .. ..2: pte 0x0000000021fd2807 pa 0x0000000087f4a000
.. .. ..3: pte 0x0000000021fd2417 pa 0x0000000087f49000
..255: pte 0x0000000021fd4001 pa 0x0000000087f50000
.. ..511: pte 0x0000000021fd3c01 pa 0x0000000087f4f000
.. .. ..510: pte 0x0000000021fd8007 pa 0x0000000087f60000
.. .. ..511: pte 0x0000000020001c0b pa 0x0000000080007000
lazytests starting
running test lazy alloc
test lazy alloc: OK
running test lazy unmap
test lazy unmap: OK
running test out of memory
test out of memory: OK
ALL TESTS PASSED
$ usertests
...
test bigdir: OK
test manywrites: OK
test badwrite: OK
test execout: OK
test diskfull: balloc: out of blocks
ialloc: no inodes
OK
test outofinodes: ialloc: no inodes
OK
ALL TESTS PASSED
$ QEMU: Terminated
```

Τερματισμός με Ctrl+A και στη συνέχεια το x.