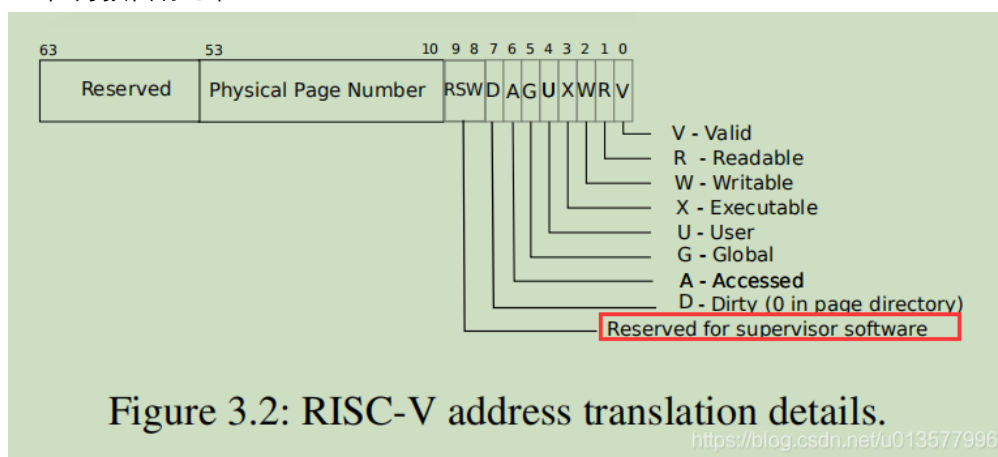


Lab5 实验报告

第一部分

这次的实验断断续续做了好久，还是没法通过所有测试，所以把我做实验的具体步骤简单说一下：

- 1) 修改 kernel/vm.c, 新增 int refNum[32768];来记录关联物理页的页表数量。32768 是根据(PHYSTOP-KERNBASE)/PGSIZE 得出。
- 2) 修改 kernel/riscv.h, 新增 PTE_COW 标志位, 参考 riscv 对 PTE 标志位定义, 第 9-10 位为预留标志位。



- 3) 修改 kernel/vm.c 的 uvmcopy(), 让进程 fork 时, 不赋值物理页, 而是 child 进程页表指向 parent 进程的物理页, 但标记要将 parent 和 child 的 PTE 都清除 PTE_W 标志位, 并添加 COW 标志位, 表明两个 PTE 指向一个物理页。
- 4) 修改 kernel/vm.c 的 mappages(), 在页表与物理页绑定时, 增加 refNum 对应元素计数。
- 5) 修改 kernel/vm.c 的 uvmunmap(), 在页表与物理页解绑时, 减少 refNum 对应元素计数, 当 refNum==1 即仅 kernel pagetable 持有时, 释放内存。
- 6) 将 kernel/vm.c 中 walk()定义在 defs.h 中。
- 7) 修改 kernel/trap.c 的 usertrap(), 引入 refNum, 在发生 page fault 时, 若该虚拟地址关联的 PTE, 表明关联的物理页是一个 COW 页, 则新申请一个物理页, 让此虚拟地址指向新物理页, 并修改 refNum 计数。
- 8) 修改 kernel/vm.c 的 copyout(), 同 kernel/trap.c 的 usertrap。

```

(4.25)
== Test    simple ==
    simple: OK
== Test    three ==
    three: OK
== Test    file ==
    file: OK
== Test    usertests ==
$ make qemu-gdb
Timeout! (300.2s)
...
                sepc=0x000000000000555e stval=0x000000000000555e
                scause 0x000000000000000d
                sepc=0x00000000800013c6 stval=0x0001010180e21a50
                panic: kerneltrap
                qemu-system-riscv64: terminating on signal 15 from pid 191882 (make)
MISSING '^ALL TESTS PASSED$'
QEMU output saved to xv6.out.usertests
== Test    usertests: copyin ==
    usertests: copyin: FAIL
    Parent failed: test_usertests
== Test    usertests: copyout ==
    usertests: copyout: FAIL
    Parent failed: test_usertests
== Test    usertests: all tests ==
    usertests: all tests: FAIL
    Parent failed: test_usertests
== Test time ==
time: OK
Score: 81/110
make: *** [Makefile:316: grade] 错误 1

```

第二部分

遇到的问题

- 有一些内存块没有回收

```

test iput: OK
test mem: OK
test pipe1: OK
test preempt: kill... wait... OK
test exitwait: OK
test rmdot: OK
test fourteen: OK
test bigfile: OK
test dirfile: OK
test iref: OK
test forktest: OK
test bigdir: OK
FAILED -- lost some free pages 32446 (out of 32448)

```

```

$ cowtest
exec cowtest failed
usertrap(): unexpected scause 0x0000000000000002 pid=2
                sepc=0x0000000000001004 stval=0x0000000000000000
usertrap(): unexpected scause 0x000000000000000c pid=1
                sepc=0x00000000000000f7c stval=0x0000000000001000
panic: init exiting

```

```
cnt = 32768
hart 2 starting
hart 1 starting
init: starting sh
panic: remap
```

解决方案

- 设计带锁的结构体来保存对物理内存的引用数
用时三天完成。

第三部分

- 这次的实验加强了我对于异常处理的熟练程度。
- 让我发觉自己对操作系统的知识还不够了解，准备再看看手册，看看能不能解决这次没解决的问题。