# Lab6 实验说明

## 指导书的说明

MOOC 上的 Lab6 部分与 pdf 指导书有轻微差别，请参照 pdf 指导书。

对于 pdf 指导书，指导书中 Lab6 的部分相对于 Lab5 发布时**没有更新**，同学们无需重新下载 pdf 指导书。

## 修复 Lab3 的 Bug

在运行 `testpipe.c` 等测试程序时，程序的输出可能会出现以下情况。`pgfault()` 函数认为我们针对一个没有 `PTE_COW` 的页面调用了 `pgfault()`。

```
fork.c:pgfault():  va:7f3fdf00
panic at fork.c:93: pgfault on non-cow page
```

修复bug：

```
void env_free(struct Env *e)
{
    Pte *pt;
    u_int pdeno, pteno, pa;

    /* Hint: Note the environment's demise.*/
    printf("[%08x] free env %08x\n", curenv ? curenv->env_id : 0, e->env_id);

    /* Hint: Flush all mapped pages in the user portion of the address space */
    for (pdeno = 0; pdeno < PDX(UTOP); pdeno++) {
        /* Hint: only look at mapped page tables. */
        if (!(e->env_pgdir[pdeno] & PTE_V)) {
            continue;
        }
        /* Hint: find the pa and va of the page table. */
        pa = PTE_ADDR(e->env_pgdir[pdeno]);
        pt = (Pte *)KADDR(pa);
        /* Hint: Unmap all PTEs in this page table. */
        for (pteno = 0; pteno <= PTX(~0); pteno++)
            if (pt[pteno] & PTE_V) {
                page_remove(e->env_pgdir, (pdeno << PDSHIFT) | (pteno << PGSHIFT));
            }
        /* Hint: free the page table itself. */
        e->env_pgdir[pdeno] = 0;
        page_decref(pa2page(pa));
        /* Hint: invalidate page table in TLB */
        tlb_invalidate(e->env_pgdir, UVPT + (pdeno << PGSHIFT));
    }
    /* Hint: free the page directory. */
    pa = e->env_cr3;
```

```
31      e->env_pgdir = 0;
32      e->env_cr3 = 0;
33      /* Hint: free the ASID */
34      asid_free(e->env_id >> (1 + LOG2NENV));
35      page_decref(pa2page(pa));
36      /* Hint: invalidate page directory in TLB. */
37      tlb_invalidate(e->env_pgdir, UVPT + (UVPT >> 10));
38      /* Hint: return the environment to the free list. */
39      e->env_status = ENV_FREE;
40      LIST_INSERT_HEAD(&env_free_list, e, env_link);
41      LIST_REMOVE(e, env_sched_link);
42  }
```

请同学们将上述代码中的 **27**、**37** 两行加入自己的 `env_free()` 的代码中。

`tlb_invalidate(e->env_pgdir, UVPT + (pdeno << PGSHIFT));`

`tlb_invalidate(e->env_pgdir, UVPT + (UVPT >> 10));`

这两句话的作用是使得页表、页目录对应的虚拟地址在 TLB 中失效。因为 ASID 号是用位图来分配的，所以后面的进程可能与前面已销毁的进程的 ASID 号相同。若不清空进程在 TLB 中的页表，则后面的进程可能会访问到前面进程的页表（例如该操作 `(* vpt)[VPN(v)]` 访问的可能是**之前**进程的页表）。