



北京航空航天大学
BEIHANG UNIVERSITY

操作系统 Lab3-1 课上测试简介



- 考试时间 14:15 ~ 16: 00
- 本次测试没有额外分支需要自己创建
- 参加本次测试需要课下测试成绩 ≥ 60
- 每次课上测试题目分为基础测试和附加测试(选做)两部分
- 通过课上测试的条件是基础测试通过（基础题成绩 ≥ 60 ）
- 通过附加测试将会给予额外加分（附加题成绩 ≥ 60 ）
- 注意，若未通过基础测试，则无论是否通过附加测试均记为未通过。



- 如要进行本地测试，应在编写完成相关函数后，在init/init.c中，调用编写的函数，编写测试代码进行测试。
- 具体示例如右侧 →

```
1  #include <asm/asm.h>
2  #include <pmap.h>
3  #include <env.h>
4  #include <printf.h>
5  #include <kclock.h>
6  #include <trap.h>
7  //int func();在以上引入的头文件中声明，在相应的源文件中实现
8  //此处命名为func仅为示例
9
10 //自己编写的测试函数测试
11 void my_test() {
12     if (func() == 0) {
13         printf("pass")
14     }
15 }
16
17 void mips_init()
18 {
19     printf("init.c:\tmips_init() is called\n");
20     mips_detect_memory();
21     mips_vm_init();
22     page_init();
23     env_init();
24
25     my_test(); //调用自己编写的测试函数
26
27     panic("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^");
28     while(1);
29     panic("init.c:\tend of mips_init() reached!");
30 }
```



Lab3-1课上基础题



Step1:创建lab3-1-exam分支

- `cd ~/学号-lab/`
- `git checkout lab3`
- `git checkout -b lab3-1-exam`



Step 2: lab3-1-exam课上基础测试

- PART1:

- 在今天的实验里我们要求你实现一个简易的 fork 函数（并不包括实际load代码段），通过给定的原始进程块(输入参数struct Env *e)生成一个新的进程控制块，并返回新进程控制块的env_id。同学们需要在 lib/env.c 和 include/env.h 中分别定义和声明 fork 函数，函数接口如下：

u_int fork(struct Env *e);

- 要求如下：
 1. 从 env_free_list 中从头申请一个新的进程控制块
 2. 新进程控制块的 env_status、env_pgdir、env_cr3、env_pri和原进程控制块保持一致。
 3. 为新进程控制块生成对应的 env_id
 4. env_parent_id 的值为原进程控制块的 env_id
 5. 返回值为新进程的env_id



Step 2: lab3-1-exam课上基础测试

• PART2:

- 本部分要求修改struct Env, 在进程控制块中增加字段 (具体增加哪些内容请自行组织) 组织起进程间的父子、兄弟关系, 并按照要求在 lib/env.c 和 include/env.h 中分别定义和声明 lab3_output 函数输出相关内容, 详情如下:
- 函数lab3_output的定义如下:

void lab3_output(u_int env_id);

- 要求输出的内容有其父进程的env_id、其第一个子进程的env_id、其前一个兄弟进程的env_id以及其后一个兄弟进程的env_id
- 所有的子进程都由fork创建, 两个进程如果是兄弟, 它们的父进程一定相同。
- 以某进程第一个子进程是指, 由该进程作为父进程使用fork创建的第一个子进程。
- 兄弟进程间的顺序即为这些进程被创建的顺序, 前一个兄弟进程为较早被创建的进程
- 需要在PART1的fork函数中进行对添加字段的修改
- 输出格式为: printf("%08x %08x %08x %08x\n", a, b, c, d);
 - 其中a, b, c, d分别为父进程的env_id、第一个子进程的env_id、前一个兄弟进程的env_id以及后一个兄弟进程的env_id
 - 如果a, b, c, d中有不存在的参数, 则输出0



Step 2: lab3-1-exam课上基础测试

• PART3:

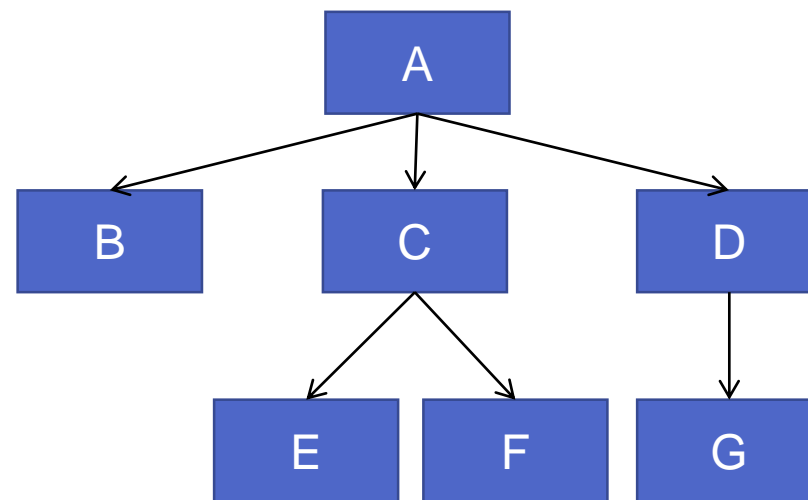
- 在PART2的基础上，在 lib/env.c 和 include/env.h 中分别定义和声明 lab3_get_sum 函数，函数的功能为：
： 给定一个进程的env_id，返回以该进程为根节点的子进程树中进程的数目（包括它本身），具体接口如下
：

```
int lab3_get_sum(u_int env_id);
```

例如右图中：

以A进程为根节点的子进程树中进程的数目为7

以C进程为根节点的子进程树中进程的数目为3





Step3:提交更改

- `cd ~/学号-lab/`
- `git add --all`
- `git commit -a -m "balabala"`
- `git push origin lab3-1-exam:lab3-1-exam`



Step4:提交结果

```
remote: /usr/local/bin/mips_4kc-10 -O gxiend1/vmlinux -N -T tools/scse0_3.10
remote:
remote: End build at 2021年 04月 18日 星期日 20:22:48 CST
remote: [ PASSED:15 ]
remote: [ TOTAL:15 ]
remote: [ You got 100 (of 100) this time. 2021年 04月 18日 星期日 20:22:58 CST ]
remote:
remote:
remote: >>>>> Collecting autotest results >>>>>
remote: 切换到一个新分支 'lab3-1-exam-result'
```

Lab3-1-exam得分说明:

- 课下强测: 10分
- PART1: 25分
- PART2: 25分
- PART3: 40分

本次测试各测试点分离，同学们可以完成一part提交一次，以观察测试情况。在测试反馈信息中有测试点通过情况的详细信息。



Lab3-1课上附加题



Step5:创建lab3-1-Extra分支 (选做)

- `cd ~/学号-lab/`
- `git checkout lab3-1-exam`
 - 注意基于lab3-1-exam创建分支
- `git checkout -b lab3-1-Extra`

Step6:完成lab3-1-Extra（选做）

- 题目要求：
- 本部分要求在 lib/env.c 和 include/env.h 中分别定义和声明 lab3_kill 函数，功能为：杀死一个进程后他的子进程成为孤儿进程由其所在进程树的根节点代表的进程收养（接管），这些孤儿进程将依次排列在根进程的子进程的尾部。测试中保证不会杀死根节点代表的进程。函数接口如下：

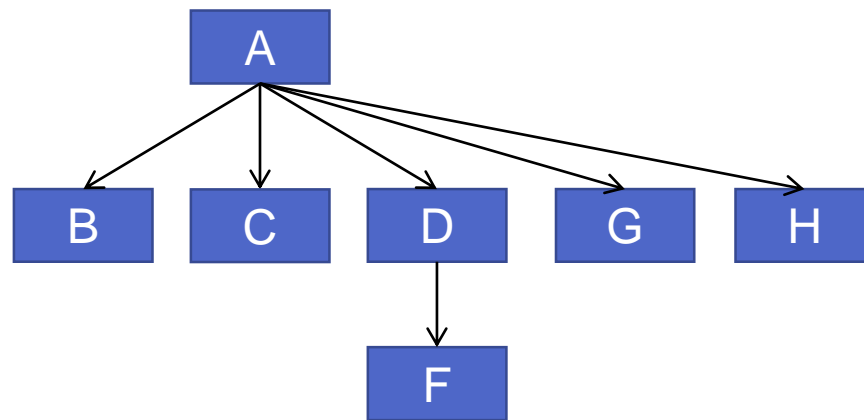
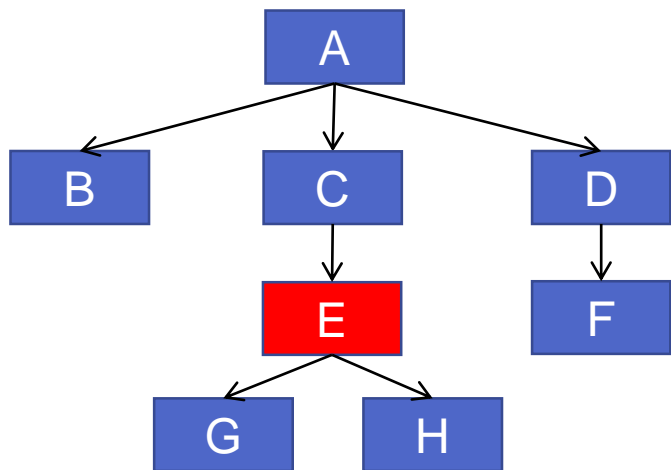
void lab3_kill(u_int env_id);

- 在完成一次删除后，基础测试PART2 lab3_output，输出的结果在新的树结构下仍应正确。
- 一颗进程树的根节点代表的进程由env_alloc创建，其余节点代表的进程皆由PART1 fork创建。
- 本部分中“杀死进程”的含义仅为释放该进程的进程控制块，需要清空进程控制块相关字段，以及其他相关处理。可以**参考**env_free（不能直接使用env_free）
- 本部分测试较强，请务必考虑完善。



Step6:完成lab3-1-Extra (选做)

- 示例：在左图的进程树结构中，杀死左图中红色的进程E，杀死E后GH排列到BCD的尾部，应得到右图中的树结构。





Step7:提交更改（选做）

- 必须通过基础测试才能获得附加题分数
- `cd ~/学号-lab/`
- `git add --all`
- `git commit -a -m "balabala..."`
- `git push origin lab3-1-Extra:lab3-1-Extra`



Step8:提交结果 (选做)

```
remote: [ PASSED:18 ]  
remote: [ TOTAL:18 ]  
remote: [ You got 100 (of 100) this time. Tue Apr 20 11:29:58 CST 2021 ]  
remote:
```




北京航空航天大学
BEIHANG UNIVERSITY

下面请同学们开始做题
有问题可以随时提问

祝实验顺利！