

1.

(1) 原题:

6. UNIX V6++中, 如果已知一个进程的 ID 号, 如何获得该进程图象可交换部分和代码段的起始地址? 如果当前进程图象的可交换部分在内存, 且需要换到盘交换区, 如何确定是否需要将代码段同时换出?

(2) 解答:

6. ① 我们已经知道进程 ID, 可以通过 PCB, 访问 process 类中的 p\_addr 元素获得进程图象可交换部分的起始地址。.  
可以借助 Text \* 找到对应的 Text 类, 并访问其 x\_caddr 获得代码段的起始地址。  
② 如果该代码段被多个进程图象所共享, 就不应换到盘交换区。

(3) 答案:

6. 【参考答案】

如果已知一个进程的 ID 号, 可根据该 ID 号在 proc 表中找到该进程的 proc 结构, 然后:

(1) 根据 proc 结构中 p\_flag 是否具有 SLOAD 标志位可判断该进程图象的可交换部分是否在内存。如果在内存, 则 p\_addr 为进程图象可交换部分在内存的起始地址, 如果在盘交换区, 则 p\_addr 为进程图象可交换部分在盘交换区上的盘块号。

(2) 根据 proc 结构中 p\_texp 找到该进程代码段的 TEXT 结构, 并根据其中 x\_ccount 的值, 判断代码段是否在内存。如果 x\_ccount $\geq$ 1, 则代码段在内存, x\_caddr 为其内存起始地址; 如果 x\_ccount=0, 则代码段在盘交换区, x\_daddr 为其在盘交换区的起始盘块号。

如果当前进程图象的可交换部分在内存, 且需要换到盘交换区, 则:

(1) 如果 x\_ccount-1=0, 则将代码段同时换出;

(2) 如果 x\_ccount-1 $>$ 0, 则将代码段不换出。

(4) 思考:

原来自己的解答比较过于笼统, 同时只是照搬课上学到和自己理解的内容, 并没有具体讨论要寻找的部分是否在内存中。比如根据 ID 号去找具体的起始地址, 就没有具体考虑是否在内存的问题, 忽略了 p\_flag 和 x\_ccount 等元素的作用。

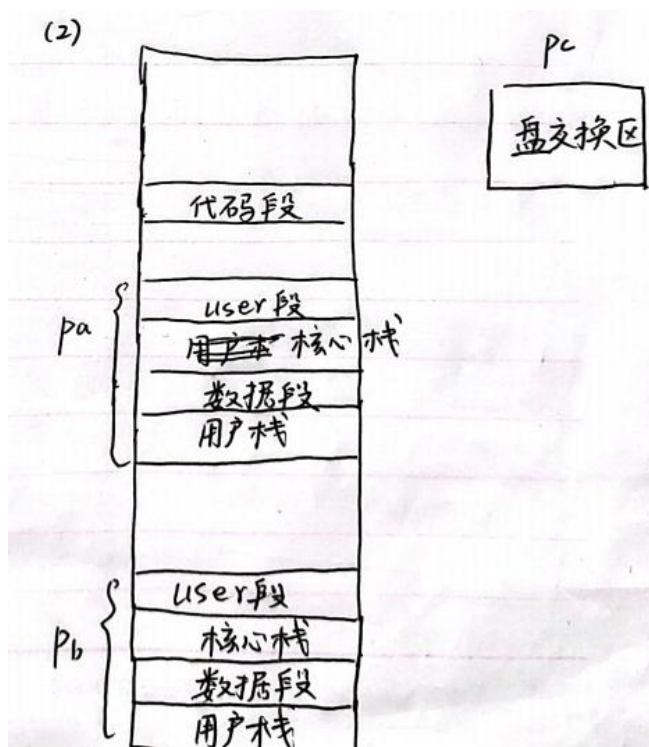
同时对于代码段是否换出也比较的笼统, 只是说出了根据共享代码的数量来确定, 但是具体的还是要看 TEXT 结构中的 x\_ccount 的元素的具体值来确定。

2.

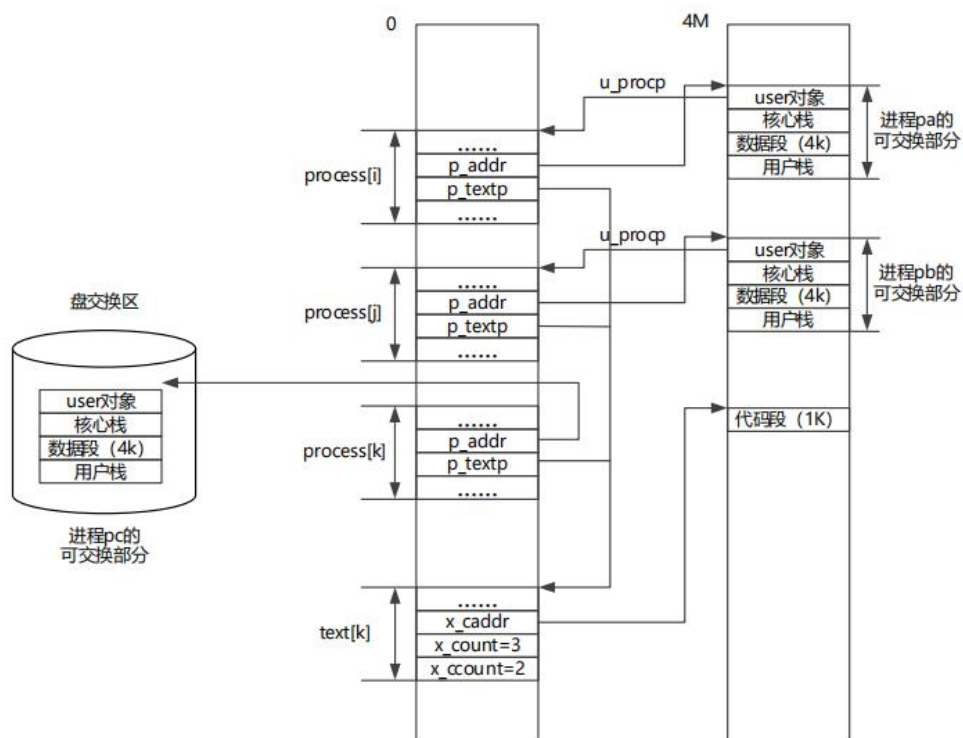
(1) 原题:

(2) 分别创建进程 pa, pb, pc 执行上述可执行文件, 其中, pa, pb 在内存, pc 在盘交换区上, 请绘制所有进程的图象。

(2) 解答:



(3) 答案:



(4) 思考:

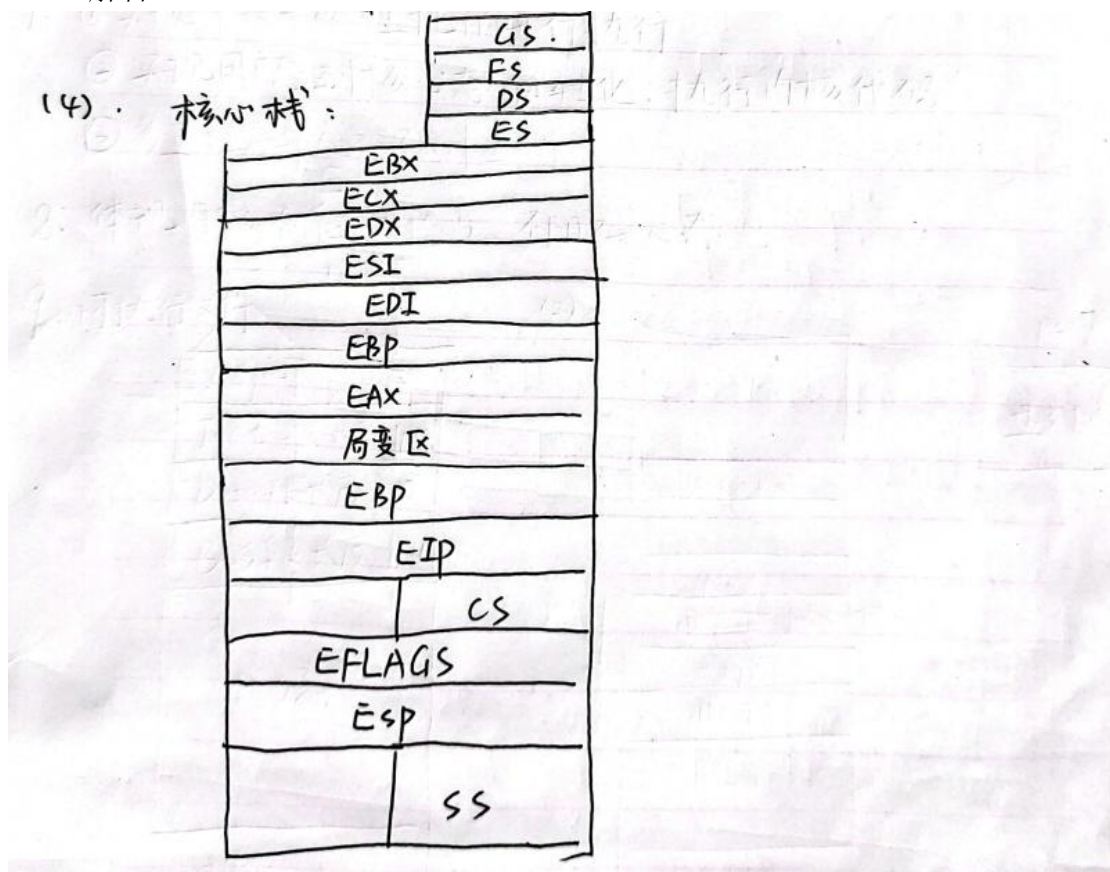
没有具体考虑到在创建新的进程的时候, 在 0-4M 的内存中, proc 表中也会出现新的 PCB, 同时其中的元素信息还与 4M 内存以外的进程图像有具体的指向关系, 在画图的过程中并没有体现出来。

3.

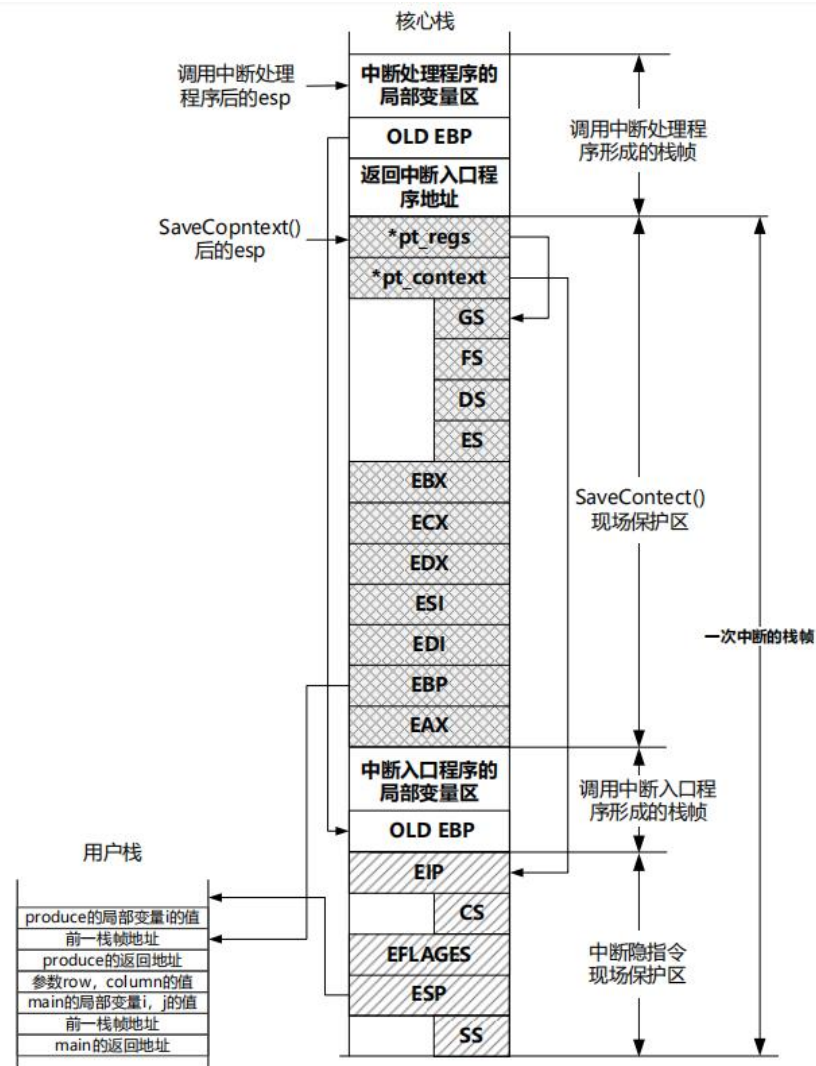
(1) 原题:

(4) 如果此时中断控制器发来一个中断请求, 且 CPU 开中断, 请绘制进程 pa 响应中断后, 核心栈的变化情况。

(2) 解答:



(3) 答案:



#### (4) 思考：

在上述的绘制过程中，没有体现出来，具体的中断处理程序的栈帧内容，没有将中断处理程序中的局变区和对应的 EBP 画出来。