

2<sup>nd</sup>

1. Describe the actions taken by a kernel to context-switch between processes. (同 1st T6)

A: 内核在进行上下文切换时, 首先将当前进程的上下文保存在内存的 PCB 中, 然后经加载调度, 下一个要执行进程在内存中 PCB 中的上下文。上下文包含 CPU 寄存器中的内容, 堆, 用户栈, 内存管理信息, 数据, 文本等。

2. A:  $pid = 0$ ,  $pid = 0$  时才能进入 A 语句, 表明当前进程为子进程。

B:  $pid = 2603$ ,  $getpid$  获得当前进程 id, 当前为子进程, 为 2603。

C:  $pid = 2603$ ,  $fork()$  函数在父进程下返回子进程 id, 为 2603。

D:  $pid = 2600$ ,  $getpid$  获得当前进程 id, 当前为父进程, id 为 2600。

## Part 2 Thread

1. Discuss the difference between user-level thread and kernel level thread.

A:

User-Level vs. kernel-level

- |                              |                                      |
|------------------------------|--------------------------------------|
| ① Managed by application     | ① Managed by kernel                  |
| ② kernel not aware of thread | ② consumes kernel resources          |
| ③ context switching cheap    | ③ context switching expensive        |
| ④ create as many as needed   | ④ number limited by kernel resources |
| ⑤ be used with care          | ⑤ simpler to use                     |

plus: 调度单位上用户线程以进程为单位, 在采用轮转(RR)算法调度时每个进程分配相同的时间片; 内核级线程则每个线程分配时间片。

2. Threads in a multithreaded process share

B. Heap memory and C. Global variables.

(don't share A. Register and D. stack memory)

3. LINEC 输出: CHILD: value = 5

LINE P 输出: PARENT: value = 0

调用fork()后, 子进程复制父进程内存空间, 此后两者独立, 父、子进程中修改不相互影响。子进程中修改 value = 5, 输出为 value = 5, 父进程中该全局变量保持不变, 仍为 0, 输出 value = 0。