

Chapter 3

3.1 Using the program shown in Figure 3.30, explain what the output will be at LINE A.

结果为：PARENT: value = 5。子进程改变的是子进程的value值，当控制权给回父进程之后，他的值保持为5。

3.5 When a process creates a new process using the fork() operation, which of the following states is shared between the parent process and the child process?

堆和栈不被共享，而子进程可以从父进程那里获得资源子集。子进程继承了父进程的权限，调度属性以及某些资源。

- a. Stack
- b. Heap
- c. Shared memory segments

只有c被共享，a、b由子进程新建

3.8 Describe the differences among short-term, medium-term, and long-term scheduling.

长期调度决定哪些进程进入到系统中，。

中期调度决定进入到系统中的进程哪些可以竞争处理器，即哪些进程可以进入到就绪队列。

短期调度决定将处理器分配给就绪队列中的哪些进程。

3.9 Describe the actions taken by a kernel to context-switch between processes.

1. 中断发生后，操作系统首先保存当前执行进程的PC和用户堆栈指针，并将控制权交给内核中断处理程序
2. 中断处理程序将其余寄存器和其他状态保存至PCB中
3. 操作系统调用调度程序确定下一个要执行的进程
4. 操作系统从下一个进程的PCB中加载其信息，将处理器还原至之前中断的状态，之后重新继续执行。

3.12 **Including the initial parent process, how many processes are created by the program shown in Figure 3.32?

每次fork一个子进程后，它都分享父进程的内存区域，所以每个fork出的子进程会接着执行循环。

一共 $1 + 1 + 2 + 4 + 8 = 16$ 个

3.13 Explain the circumstances under which the line of code marked `printf("LINE J")` in Figure 3.33 will be reached.

`exec1p()` 函数如果执行成功则函数不会返回, 执行失败则直接返回-1, 失败原因存于`errno` 中.

因此如果调用 `exec1p()` 函数执行成功，则不会到达“LINE J”，如果 `exec1p()` 执行失败，到达“LINE J”

3.14 Using the program in Figure 3.34, identify the values of `pid` at lines A, B, C, and D. (Assume that the actual pids of the parent and child are 2600 and 2603, respectively.)

`pid = fork()` 有两个返回值，在父进程中返回子进程id，在子进程中返回0

A: 0

B: 2603

C: 2603

D: 2600

3.17 Using the program shown in Figure 3.35, explain what the output will be at lines X and Y.

输出如下：

```
X: CHILD: 0 CHILD: -1 CHILD: -4 CHILD: -9 CHILD: -16
Y: PARENT: 0 PARENT: 1 PARENT: 2 PARENT: 3 PARENT: 4
```

因为子进程是父进程的副本，因此子对象所做的任何更改都将发生在其数据副本中，而不会反映在父对象中。因此，子对象在X行输出的值是0、-1、-4、-9、-16。父项在Y行的输出值为0、1、2、3、4