1. fork()函数

fork：创建一个和当前映像一样的进程可以通过fork()系统调用

#include<sys/types.h>

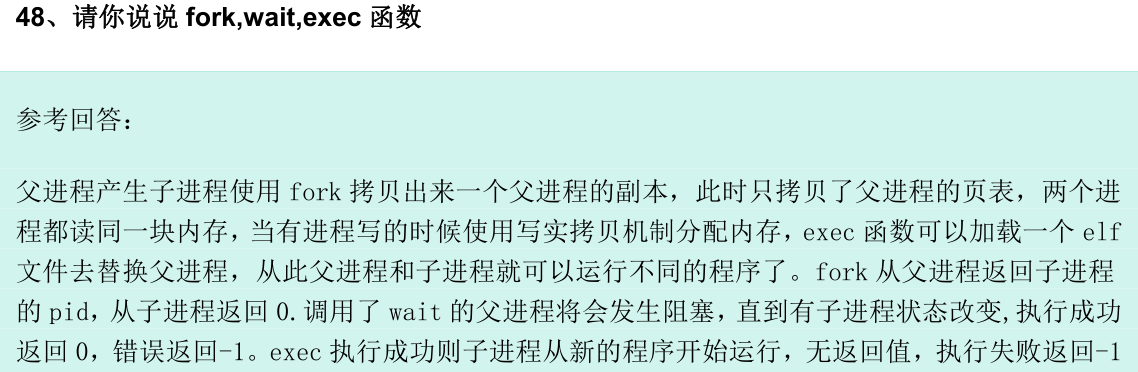
#include<unistd.h>

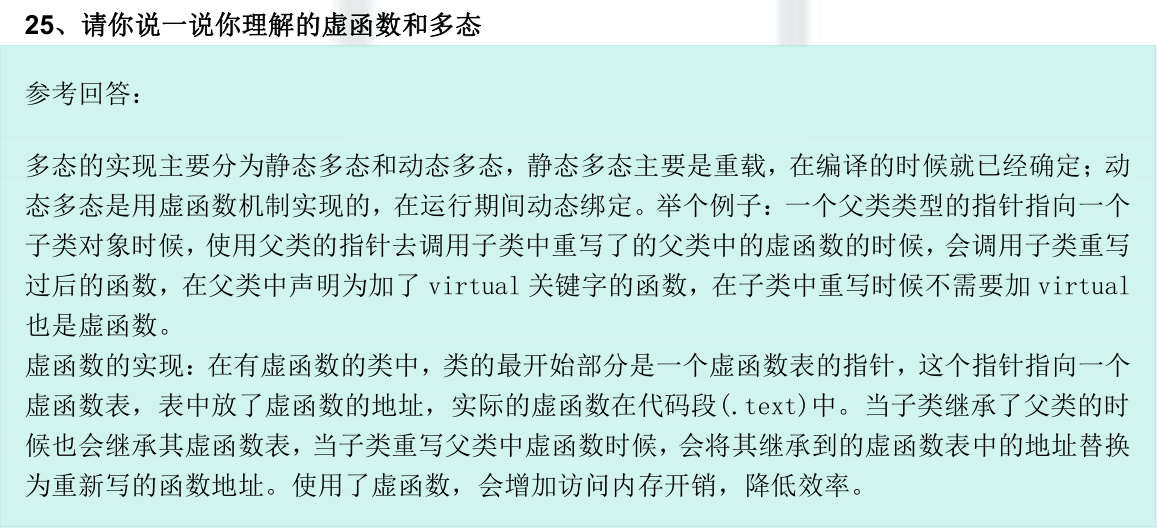
pid\_t fork(void);

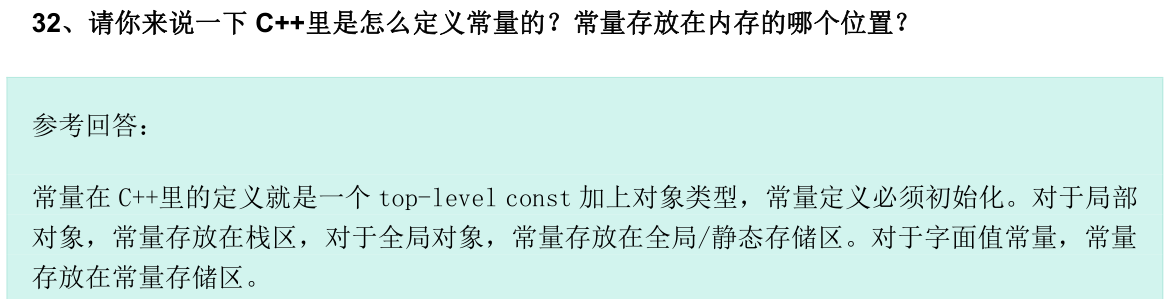
成功调用fork()会创建一个新的进程，它几乎与调用fork()的进程一摸一样，这两个进程都会继续运行。在子进程中，成功的fork()调用会返回0，在父进程中fork()返回子进程的pid。如果出现错误，fork()会返回一个负值。

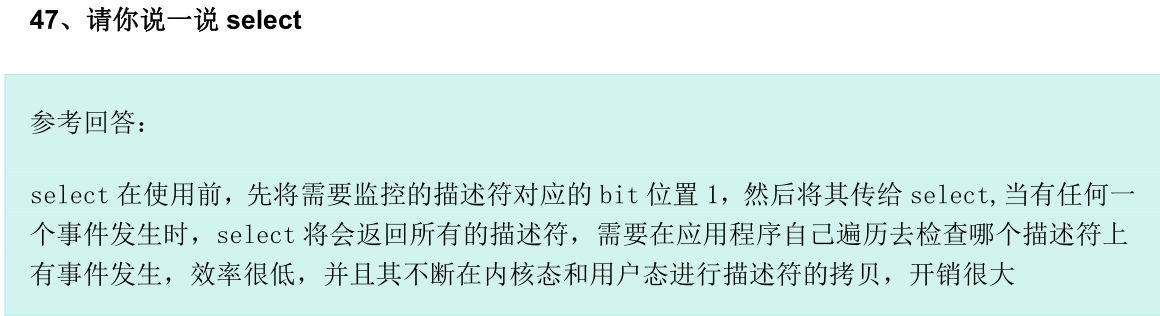
最常见的fork()用法是创建一个新的进程，然后使用exec()载入二进制映像，替换当前进程的映像，这种情况下，派生fork了新的进程，而这个子进程会执行一个新的二进制可执行文件的映像。这种“派生加执行”的方式是很常见的。

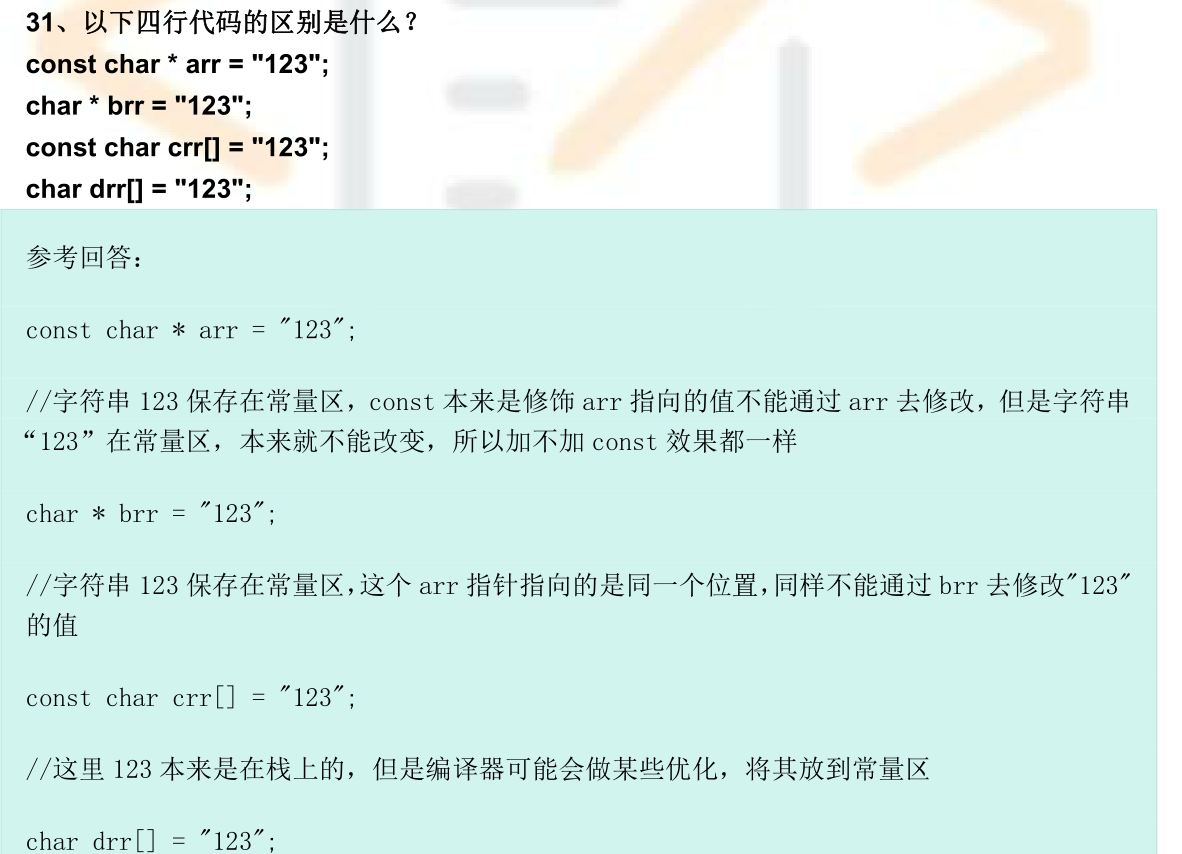
在早起的unix系统中，创建进程比较原始。当调用fork时，内核会把所有的内部数据结构复制一份，复制进程的页表项，然后把父进程的地址空间中的内容逐页的复制到子进程的地址空间中。但从内核角度来说，逐页的复制方式时十分耗时的，现代的unix系统采取了更多的优化，例如linux，采用了写时复制的方法，而不是对父进程空间进程整体复制。











//字符串123保存在栈区，可以通过drr修改

