

ICS 第十章

1. 假设缓冲区足够大，且 stdout 只有在关闭文件、换行与 fflush 的情况下才会刷新缓冲区。程序运行过程中的所有系统调用均成功。

(1)	(2)	(3)
<pre>int main() {     printf("a");     fork();     printf("b");     fork();     printf("c");     return 0; }</pre>	<pre>int main() {     write(1, "a", 1);     fork();     write(1, "b", 1);     fork();     write(1, "c", 1);     return 0; }</pre>	<pre>int main() {     printf("a");     fork();     write(1, "b", 1);     fork();     write(1, "c", 1);     return 0; }</pre>

对于(1)号程序，写出它的一个可能的输出：\_\_\_\_\_。这个可能的输出是唯一的吗？\_\_\_\_\_。

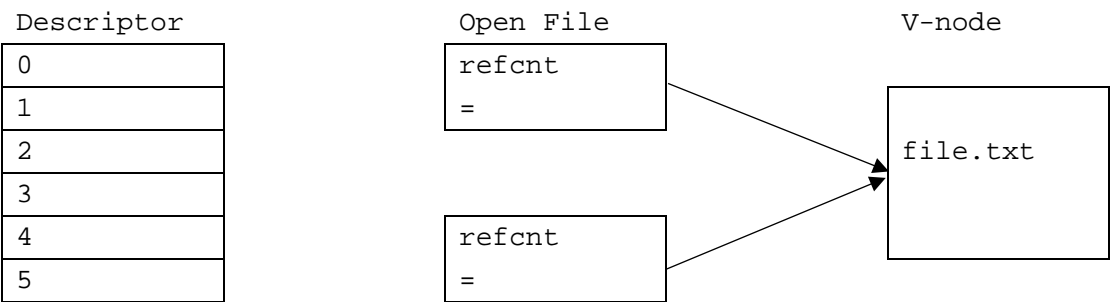
对于(2)号程序，它的输出中包含\_\_\_\_\_个 a，\_\_\_\_\_个 b，\_\_\_\_\_个 c。输出的第一个字符一定是\_\_\_\_\_。

对于(3)号程序，它的输出中包含\_\_\_\_\_个 a，\_\_\_\_\_个 b，\_\_\_\_\_个 c。输出的第一个字符一定是\_\_\_\_\_。

2. 假设磁盘上有空文件 file.txt。程序运行过程中的所有系统调用均成功。

```
int main() {
    int fd1 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
    int fd2 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
    printf("%d %d\n", fd1, fd2);
    write(fd1, "123", 3);
    write(fd2, "45", 2);
    close(fd1); close(fd2);
    return 0;
}
```

(1) 程序关闭 fd1 前，画出 LINUX 三级表结构。填写 Open File 表中的 refcnt。

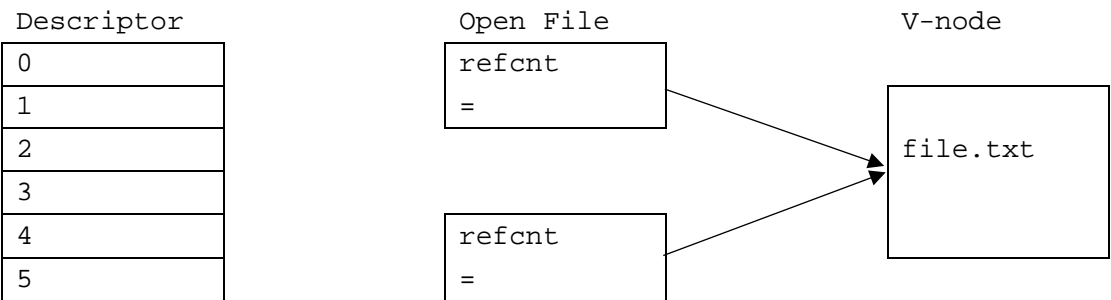


(2) 程序结束时，标准输出上的内容是\_\_\_\_\_，file.txt 中的内容是\_\_\_\_\_。

3. 假设磁盘上有空文件 file.txt。程序运行过程中的所有系统调用均成功。

```
int main() {
    int fd1 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
    int fd2 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
    dup2(fd2, fd1);
    printf("%d %d\n", fd1, fd2);
    write(fd1, "123", 3);
    write(fd2, "45", 2);
    close(fd1); close(fd2); return 0;
}
```

(1) 程序关闭 fd1 前，画出 LINUX 三级表结构。填写 Open File 表中的 refcnt。

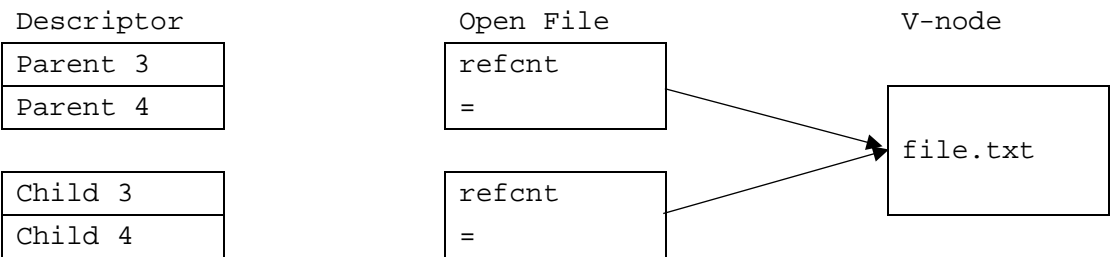


(2) 程序结束时，标准输出上的内容是\_\_\_\_\_， file.txt 中的内容是\_\_\_\_\_。

4. 假设磁盘上有空文件 file.txt。程序运行过程中的所有系统调用均成功。缓冲区足够大，且 stdout 只有在关闭文件、换行与 fflush 的情况下才会刷新缓冲区。

```
int main() {
    pid_t pid; int child_status;
    int fd1 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
    if ((pid = fork()) > 0) { // Parent
        printf("P:%d ", fd1);
        write(fd1, "123", 3);
        waitpid(pid, &child_status, 0);
    } else { // Child
        printf("C:%d ", fd1);
        write(fd1, "45", 2);
    }
    close(fd1); return 0;
}
```

(1) 子进程关闭 fd1 前，画出 LINUX 三级表结构。填写 Open File 表中的 refcnt。



(2) 程序结束时，标准输出上的内容是\_\_\_\_\_， file.txt 中的内容是\_\_\_\_\_。