1.

#include <stdio.h>

main()

{

int p1, p2;

while(p1=fork()==-1); /\* 创建子进程p1 \*/

if (p1==0) /\* 子进程创建成功 \*/

putchar('b');

else

{

while(p2=fork()==-1); /\* 创建另一个子进程 \*/

if (p2==0) /\* 子进程创建成功 \*/

putchar('c');

else

putchar('a'); /\* 父进程执行 \*/

}

}

结果与分析：多次执行的话，a/b/c的显示顺序不一定，取决于进程的调度时机。

2.

#include <stdio.h>

main()

{

int p1, p2, i;

while(p1=fork())==-1);

if (p1==0)

for (i=0;i<50;i++)

printf("child %d\n", i);

else

{

while(p2=fork())==-1);

if (p2==0)

for(i=0;i<50;i++)

printf("son %d\n");

else

for (i=0;i<50;i++)

printf("daughter %d\n",i);

}

结果与分析：

子进程1

子进程2

父进程

成功显示预设的话，其中顺序与进程的调度机制有关。

3.

#include <stdio.h>

#include <unistd.h>

main()

{

int p1, p2, i;

while((p1=fork())==-1);

if (p1==0)

{

lock(1,1,0);

for (i=0;i<50;i++)

printf("child %d\n", i);

lock(1,0,0);

}

else

{

while(p2=fork())==-1);

if (p2==0)

{

lock(1,1,0);

for(i=0;i<50;i++)

printf("son %d\n");

lock(1,0,0);

}

else

{

lock(1,1,0);

for (i=0;i<50;i++)

printf("daughter %d\n",i);

lock(1,0,0);

}

}

}

结果与分析：大致与未上锁的输出结果相同，也是随着执行时间不同，输出结果的顺序有所不同。