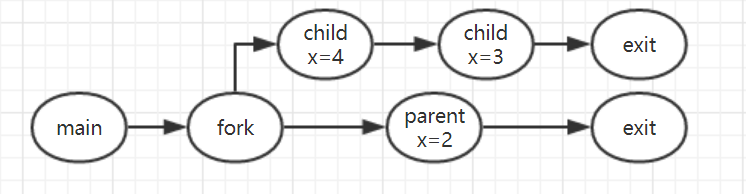
# 8.9

|  |  |
| --- | --- |
| **进程对** | **并发地？** |
| **AB** | **否** |
| **AC** | **是** |
| **AD** | **是** |
| **BC** | **是** |
| **BD** | **是** |
| **CD** | **是** |

# 8.13

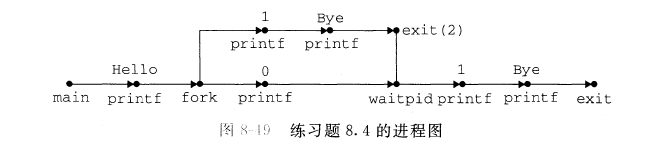


**x=4**

**x=3**

**x=2**

# 8.17

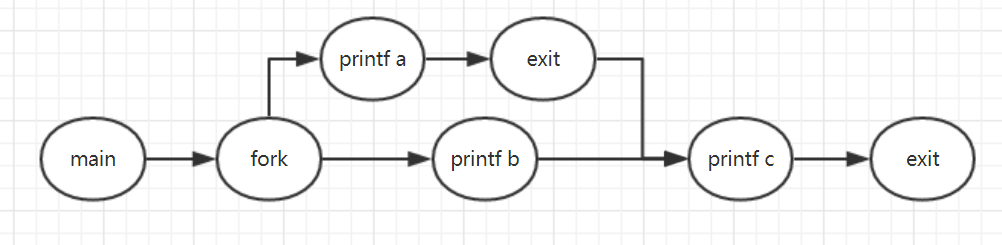


**1．Hello，0，1，Bye，2，Bye**

**2．Hello，1，0，Bye，2，Bye**

**3．Hello，1，Bye，0，2，Bye**

# 8.21



**Waitpid(-1,NULL,0)会等待子进程返回之后，再继续进行。所以c一定在a，b之后。**

**答案为abc，或bac**

# 8.25

**用非本地跳转和alarm来解决问题。**

**注意tfgets\_handler本身即是信号处理程序，这里不能用longjmp，和setjmp，而要用siglongjmp和sigsetjmp**

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <unistd.h>

#include <signal.h>

#include <setjmp.h>

sigjmp\_buf env;

void tfgets\_handler(int sig)

{

signal(SIGALRM, SIG\_DFL);

siglongjmp(env, 1);

}

char \*tfgets(char \*buf, int bufsize, FILE \*stream)

{

signal(SIGALRM, tfgets\_handler); **//设置SIGALRM的处理函数**

alarm(5); **//5秒后发送一个SIGALRM**

int i = sigsetjmp(env, 1);

if(i== 0) return fgets(buf, bufsize, stream); **//如果SIGALRM没有出现，i=0，返回输入**

else return NULL; **//如果SIGALRM出现，siglongjmp跳转到sigsetjmp**

} **//处使得i=1；**

int main() {

char buf[50];

char\* input = tfgets(buf, 50, stdin);

if (input == NULL) {

printf("没有输入\n");

} else {

printf("输入了:%s", input);

}

return 0;

}

**测试结果：**

