

- 8.3 我们知道序列 acbc、abcc 和 bacc 是可能的，因为它们对应有进程图的拓扑排序(图 8-48)。而像 bcac 和 cbca 这样的序列不对应有任何拓扑排序，因此它们是不可行的。

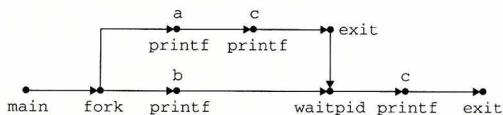


图 8-48 练习题 8.3 的进程图

- 8.4 A. 只简单地计算进程图(图 8-49)中 printf 顶点的个数就能确定输出行数。在这里，有 6 个这样的顶点，因此程序会打印 6 行输出。  
B. 任何对应有进程图的拓扑排序的输出序列都是可能的。例如：Hello、1、0、Bye、2、Bye 是可能的。

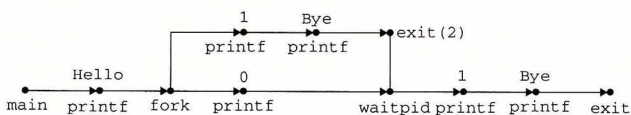


图 8-49 练习题 8.4 的进程图

- 8.5 code/ecf/snooze.c

```
1 unsigned int snooze(unsigned int secs) {
2     unsigned int rc = sleep(secs);
3
4     printf("Slept for %d of %d secs.\n", secs-rc, secs);
5     return rc;
6 }
```

code/ecf/snooze.c

- 8.6 code/ecf/myecho.c

```
1 #include "csapp.h"
2
3 int main(int argc, char *argv[], char *envp[])
4 {
5     int i;
6
7     printf("Command-line arguments:\n");
8     for (i=0; argv[i] != NULL; i++)
9         printf("    argv[%2d]: %s\n", i, argv[i]);
10
11     printf("\n");
12     printf("Environment variables:\n");
13     for (i=0; envp[i] != NULL; i++)
14         printf("    envp[%2d]: %s\n", i, envp[i]);
15
16     exit(0);
17 }
```

code/ecf/myecho.c

- 8.7 只要休眠进程收到一个未被忽略的信号，sleep 函数就会提前返回。但是，因为收到一个 SIGINT 信号的默认行为就是终止进程(图 8-26)，我们必须设置一个 SIGINT 处理程序来允许 sleep 函数返回。处理程序简单地捕获 SIGNAL，并将控制返回给 sleep 函数，该函数会立即返回。

code/ecf/snooze.c

```
1 #include "csapp.h"
2
```