



操作系统原理

Operating Systems Principles

陈鹏飞
计算机学院



作业-3

1、 下图程序在LINE A处的输出是什么？为什么会有这样的输出？

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int value = 5;

int main()
{
    pid_t pid;

    pid = fork();

    if (pid == 0) { /* child process */
        value += 15;
        return 0;
    }
    else if (pid > 0) { /* parent process */
        wait(NULL);
        printf("PARENT: value = %d",value); /* LINE A */
        return 0;
    }
}
```



作业-3

2、下图程序运行过程中一共会出现多少个进程（包括主进程）？

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    /* fork a child process */
    fork();

    /* fork another child process */
    fork();

    /* and fork another */
    fork();

    return 0;
}
```



作业-3

3、当一个进程使用fork () 操作创建一个新进程时 下列状态之一在父进程和子进程之间共享:

- a. 栈
- b. 堆
- c. 共享内存段

4、描述内核对进程之间的上下文切换所采取的操作。

5、举一个普通管道比命名管道更适合的情况的例子, 以及一个命名管道比普通管道更适合情况的例子。



作业-3

6、假设父和子进程的实际pid分别为2600和2603，请给出A、B、C、D处的输出

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
    pid_t pid, pid1;

    /* fork a child process */
    pid = fork();

    if (pid < 0) { /* error occurred */
        fprintf(stderr, "Fork Failed");
        return 1;
    }
    else if (pid == 0) { /* child process */
        pid1 = getpid();
        printf("child: pid = %d",pid); /* A */
        printf("child: pid1 = %d",pid1); /* B */
    }
    else { /* parent process */
        pid1 = getpid();
        printf("parent: pid = %d",pid); /* C */
        printf("parent: pid1 = %d",pid1); /* D */
        wait(NULL);
    }

    return 0;
}
```
