

操作系统原理 (CS040047X)

作业 #3 (2020 年 10 月 16 日)

教师: Rui Li, 刘彩苹

学号及姓名: _____

温馨提示: 请于 2020 年 10 月 30 日前提交作业. **严禁抄袭或拷贝作业。**

作业采用电子版 WORD (或 PDF) 形式, 文件统一用序号 + 姓名 + 第 * 次作业的方式命名, 例如 03 黄愉情第一次作业.docx, 02 王易第一次作业.docx 等, 请各专业的课代表在指定日期前收齐作业并发送至邮箱 liucaiping@hnu.edu.cn。

题目 1

使用下图所示的程序, 说明 LINE A 的输出是什么。为什么?

```
#include < sys/types.h >
#include < stdio.h >
#include < unistd.h >
int value = 5;
int main(){
    pid_t pid;
    pid= fork();
    if (pid == 0) {
        value += 15;
    }else if(pid > 0) {
        wait(NULL);
        printf("PARENT: value = %d", value); /*LINE A*/
        exit();
    }
}
```

解答:**题目 2**

下面设计的优点和缺点分别是什么? 分别从操作系统层面和用户层面来阐述。

- 同步和异步通信

- 自动和显式缓冲
- 复制传送和引用传送
- 固定大小和可变大小消息

解答:

题目 3

Fibonacci 序列是一组数: 0, 1, 1, 2, 3, 5, 8, ..., 通常它可以表示为:

$$fib_0 = 0$$

$$fib_1 = 1$$

$$fib_n = fib_{n-1} + fib_{n-2}$$

使用系统调用 `fork()` 编写一个 C 程序, 使其在子程序中生成 Fibonacci 序列, 序列的号码将在命令行中提供。例如, 如果提供的是 5, Fibonacci 序列中的前 5 个数将由子进程输出。退出程序前, 父进程调用 `wait()` 调用来等待子进程结束。执行必要的错误检查以保证不会接受命令行传递来的负数号码。

解答:

题目 3

请检索文献了解某一特定操作系统 (如 Solaris, Windows 等) 所提供的进程状态及其可能的状态转换关系, 并与基本的进程状态转换图进行比较。

解答:

题目 4

请详细描述上下文的切换过程，并谈谈你对上下文切换的作用理解。

解答：

题目 5

请简述你对进程的理解，并分析进程与程序的区别和联系。

解答：