**北京邮电大学软件学院**

**2018学年第1学期实验报告**

**课程名称： 操作系统**

**实验名称： 进程管理**

**实验完成人：**

**姓名：周子格**

**学号：2016211978**

**成绩：**\_\_\_\_\_\_\_\_

**指导教师：**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**日 期： 2018 年 3 月 1 日**

1. **实验目的**

（1）理解进程的概念，明确进程和程序的区别。

（2）理解并发执行的实质。

（3）掌握进程的睡眠、同步、撤销等进程控制方法。

1. **实验内容**

2.1. 进程的创建。

① 编写一段源程序，使系统调用fork()创建两个子进程，当此程序运行时，在系统中 有一个父进程和两个子进程活动。让每一个进程在屏幕上显示一个字符：父进程显示字符 “a”；子进程分别显示字符“b”和字符“c”。试观察记录屏幕上的显示结果，并分析原 因。

② 修改已编写的程序，将每个进程输出一个字符改为每个进程输出一句话，在观察程序执行时屏幕出现的现象，并分析原因。

2.2. 进程的控制 ① 用fork()创建一个进程，再调用exec()用新的程序替换该子进程的内容。 ② 利用wait()来控制进程执行顺序。

1. **实验环境**

Linux centos

1. **实验过程描述**

在这里分别以文字和流程图的方式画出你的实验的几个阶段。

1. 实验内容分析

Lab1\_1\_1:在父进程中，用Fork()创建子进程1.因为当fork()<0时，说明创建不成功，所以在while循环中创建，知道创建成功才能跳出循环。若pid==0，说明当前是子进程在活动，打印字符‘a’。若pid>0，说明在父进程中，不执行打印字符的活动。

Lab1\_1\_2:函数printf（）输出的字符串之间不会被中断，因此字符串内部的顺序输出时是不变的

Lab1\_2\_1: 在使用exec函数族时，一定要加上错误判断语句。因为exec很容易执行失败

Lab1\_2\_2: int wait(int \*status)

函数功能是：父进程一旦调用了wait就立即阻塞自己，由wait自动分析是否当前进程的某个子进程已经退出，如果让它找到了这样一个已经变成僵尸的子进程，wait就会收集这个子进程的信息，并把它彻底销毁后返回；如果没有找到这样一个子进程，wait就会一直阻塞在这里，直到有一个出现为止。

若当前进程为父进程，则等待。若是子进程，则完成任务，完成后exit(5)。子进程结束后开始父进程的执行。调用函数WEXITSTATUS(status) 当WIFEXITED返回非零值时，我们可以用这个宏来提取子进程的返回值，如果子进程调用exit(5)退出，WEXITSTATUS(status) 就会返回5。

5 相关原理介绍

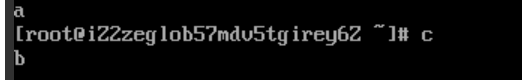
6 程序设计、调试

7 程序运行、跟踪和成果收集

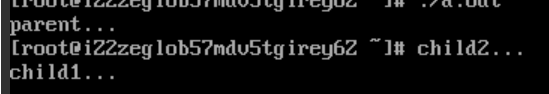
8 实验总结

1. **实验结果**

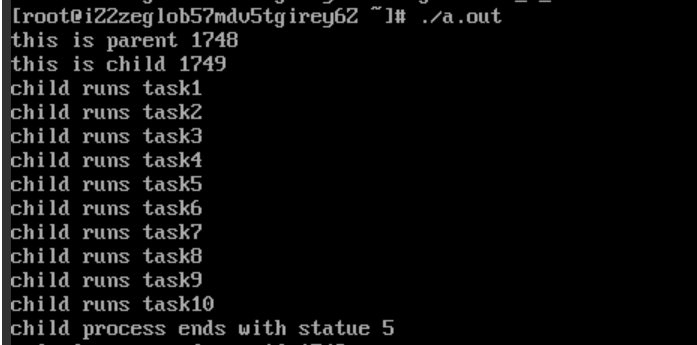
**Lab1\_1**



**Lab1\_2:**

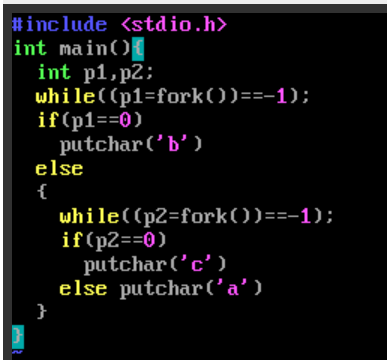


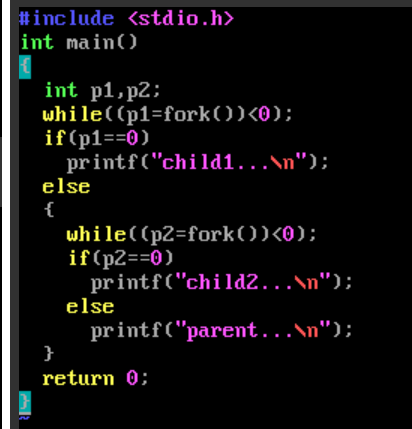
**Lab1\_2\_2:**

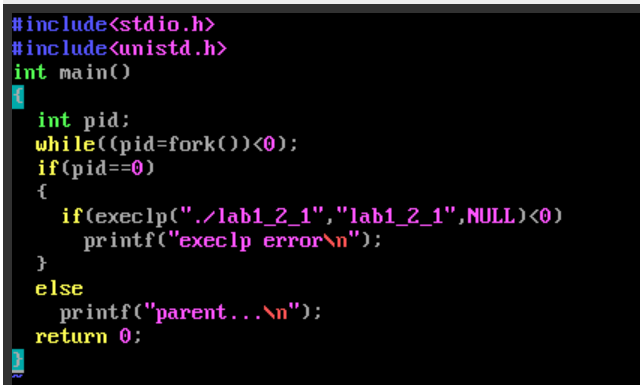


1. **附件**

**6.1 附件1：源代码**







**6.2** 