

实验2——fork () 系统调用

1.实验内容

- 任务一：fork的基本使用

进程的创建实验。使用学会使用系统调用函数fork()创建一个子进程，并且为了显示进程之间的调用和运行关系，使用在子父进程中打印字符串的方式显式的指明进程的运行关系。

父进程：“I am a Parent”

子进程：“I am a Child”

使用运行脚本“test.sh”将程序运行100次观察结果。

随后，将fork.c程序中的wait(NULL),即等待子进程结束的代码去掉，重新运行脚本“test.sh”，观察运行结果。

针对于本实验中出现的结果请在实验报告中予以解释和说明。

- 任务二：深入理解fork创建的子进程与父进程的关系

观察给定的任务二代码，先思考会输出几个“-”，随后运行程序观察程序实际输出情况。解释相关实验现象，并根据对程序实际输出的分析画出子父进程间的层级关系。（tips：缓冲区、fork创建子进程的主要过程）

2.实验相关代码

任务一

- fork.c

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<unistd.h>
4  #include<sys/wait.h>
5  int main(void)
6  {
7      pid_t pid;
8      pid = fork();
9      if(pid < 0){
10         fprintf(stderr,"Fork Failed");
11         exit(-1);
12     }
13     else if(pid == 0){
14         printf("I am a child\n");
15     }
16     else{
17         wait(NULL);
18         printf("I am a parent\n");
19         exit(0);
20     }
21 }
```

- test.sh

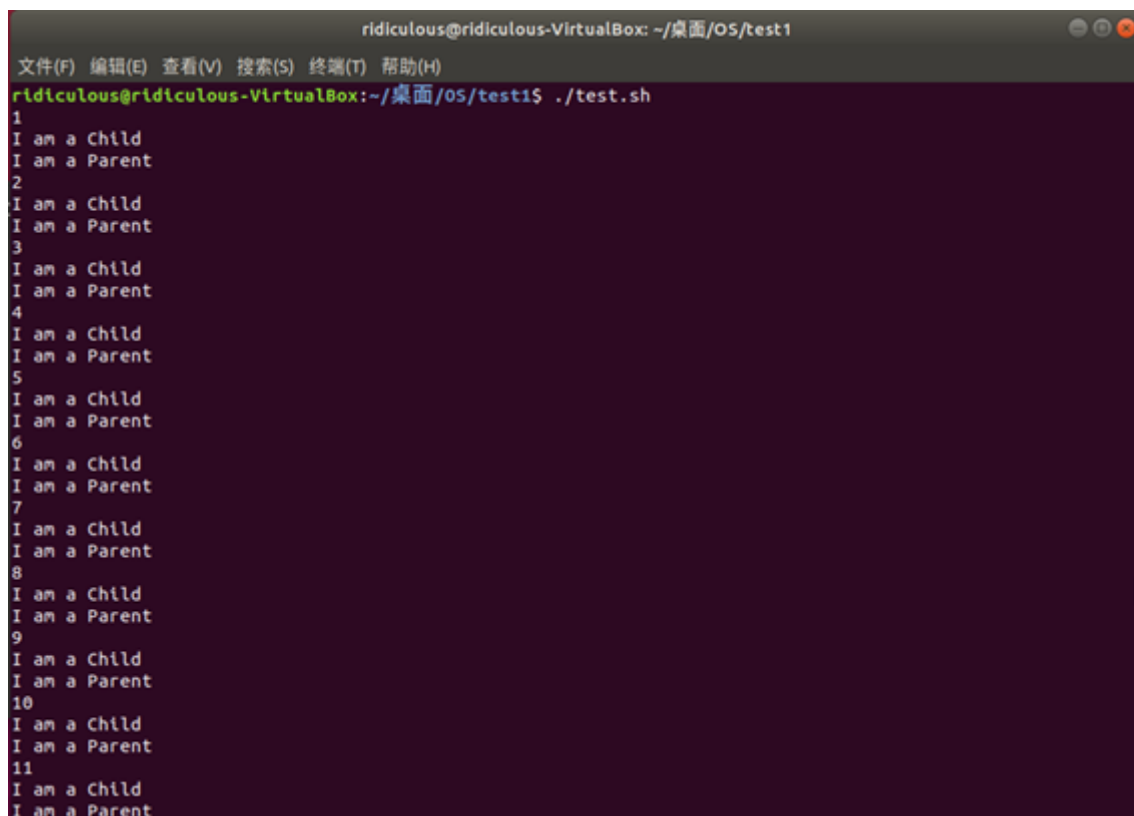
```
1  #!/bin/bash
2  for((i=1;$i<=100;i++));
3  do
4  echo $i
5  ./fork
6  done
```

任务二

```
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <unistd.h>
4
5  int main(void)
6  {
7      int i;
8      for(i=0; i<2; i++){
9          fork();
10         printf("-");
11     }
12     wait(NULL);
13     wait(NULL);
14     return 0;
15 }
```

3.实验结果示例

任务一



```
ridiculous@ridiculous-VirtualBox: ~/桌面/OS/test1
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ridiculous@ridiculous-VirtualBox:~/桌面/OS/test1$ ./test.sh
1
I am a Child
I am a Parent
2
I am a Child
I am a Parent
3
I am a Child
I am a Parent
4
I am a Child
I am a Parent
5
I am a Child
I am a Parent
6
I am a Child
I am a Parent
7
I am a Child
I am a Parent
8
I am a Child
I am a Parent
9
I am a Child
I am a Parent
10
I am a Child
I am a Parent
11
I am a Child
I am a Parent
```

有 wait(NULL)的情况

```
ridiculous@ridiculous-VirtualBox: ~/桌面/OS/test1
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
I am a Parent
I am a Child
58
I am a Parent
I am a Child
59
I am a Parent
I am a Child
60
I am a Child
I am a Parent
61
I am a Parent
I am a Child
62
I am a Parent
I am a Child
63
I am a Parent
I am a Child
64
I am a Parent
I am a Child
65
```

没有 wait(NULL)的情况