

# 操作系统实验报告

顾芄骐 20079100001

2023 年 4 月 10 日

## 1 进程的建立

- 实验目的：学会通过基本的 Windows 或者 Linux 进程控制函数，由父进程创建子进程，并实现父子进程协同工作。
- 实验软件环境：Ubuntu22.04 & GNU-C++17
- 实验内容：创建两个进程，让子进程读取一个文件，父进程等待子进程读取完文件后继续执行，实现进程协同工作。进程协同工作就是协调好两个进程，使之安排好先后次序并以此执行，可以用等待函数来实现这一点。当需要等待子进程运行结束时，可在父进程中调用等待函数。

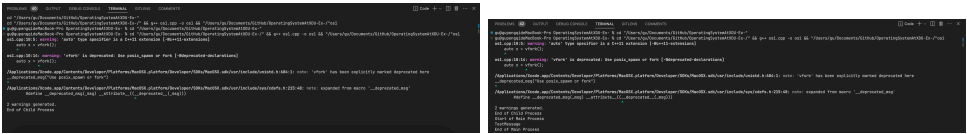
### 1.1 代码实现

在“*os1.cpp*”中写如下代码，并创建名为“*os1.in*”的文件，在后者中写入且仅写入“TextMessage”作为测试输出内容。

```
1 #include<fstream>
2 #include<unistd.h>
3 #include<stdio.h>
4 #include<iostream>
5 using namespace std;
6 int main(){
7     int flag = 0;
8     string answer = "";
9     auto x = vfork();
10    if (!x) {
11        ifstream in("os1.in");
12        in >> answer;
13        in.close();
14        flag = 1;
15        cout << "End of Child Process" << endl;
16    } else {
17        while (!flag);
18        cout << "Start of Main Process" << endl;
19        cout << answer << endl;
20        cout << "End of Main Process" << endl;
21    }
22    return EXIT_SUCCESS;
23 }
```

### 1.2 实验结果

运行上述代码，可得到如下左图所示的实验结果。片刻之后，终端显示如右所示的结果。



1.3 实验结果分析

在 1.1 所示代码中，子进程读取了文件“os1.in”；读取完成后，父进程继续执行，并显示了开始、结束和子进程所读取的文件内容。需要注意的是，使用 **vfork** 时修改静态区变量是未定义行为，会引发“栈溢出（stack smashing）”。