

浙江工业大学

操作系统原理实验报告

(2019 级)



实验题目 实验一-进程创建

学生姓名 马佳敏

学生学号 201906021316

学科(专业) 软件工程 1901 班

所在学院 计算机科学与技术学院

提交日期 2021 年 10 月 14 日

实验一、进程创建

1.1 实验目的

利用 Windows 提供的 API 函数，编写程序，实现进程的创建和终止（如创建写字板进程及终止该进程），加深对操作系统进程概念的理解，观察操作系统进程运行的动态性能，获得包含多进程的应用程序编程经验。

1.2 实验内容及要求

- 进程的创建和终止。编写一段程序，可以创建一个进程，并终止当前创建的进程。试观察记录程序执行的结果，并分析原因。
- 利用 VC 实现上述程序设计和调试操作，对于进程创建的成功与否、终止进程操作的成功与否提供一定的提示框。
- 通过阅读和分析实验程序，学习创建进程、观察进程和终止进程的程序设计方法

1.3 实验技术

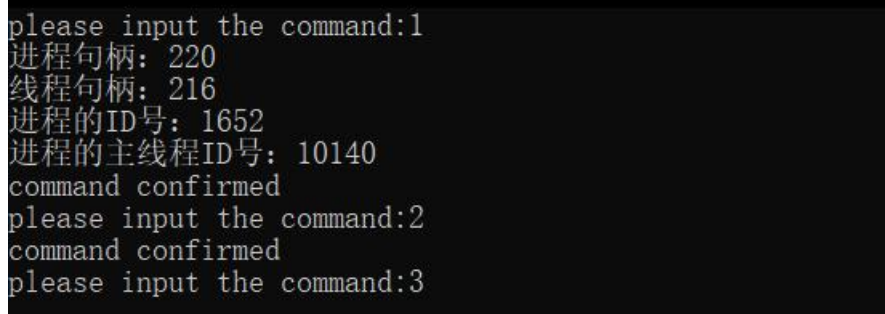
- 1) 初始化进程对象：CreateProcess() API 函数
- 2) 终止进程：ExitProcess() 或 TerminateProcess() API 函数

1.4 实验步骤

- 1) 使用 CreateProcess() API 函数初始化进程对象
- 2) 输出进程对象的一些参数，如进程 ID、进程的主线程 ID
- 3) 使用 ExitProcess() 或 TerminateProcess() API 函数终止进程

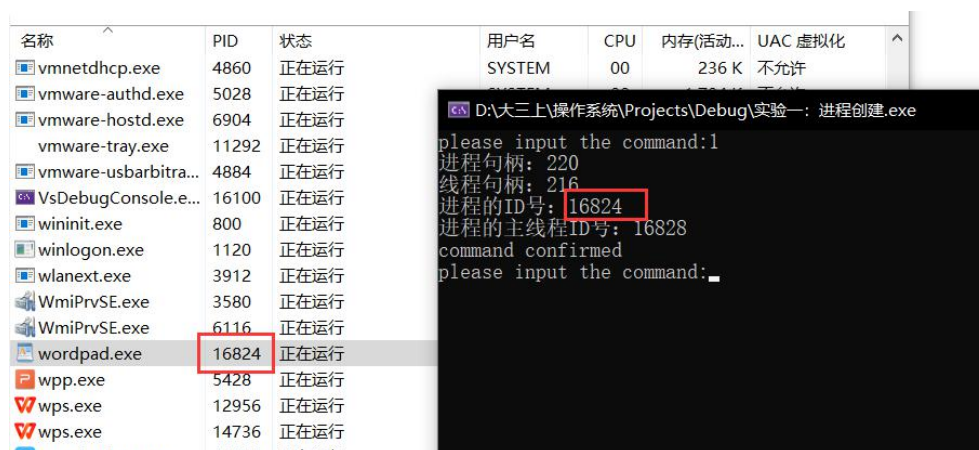
1.5 实验结果及分析说明

- 1) 打开程序，在输入窗口输入 1 可创建写字板进程，打开写字板；输入 2 可终止进程，关闭写字板；输入 3 可退出程序。结果如下图所示：



```
please input the command:1
进程句柄: 220
线程句柄: 216
进程的ID号: 1652
进程的主线程ID号: 10140
command confirmed
please input the command:2
command confirmed
please input the command:3
```

- 2) 创建进程时会输出该进程的进程句柄、线程句柄、进程 ID 号和主线程的 ID 号。进程 ID 与任务管理器中的 PID 一致，如下图所示：



3) 创建进程使用的是 `CreateProcess()` API 函数，其中第一个参数 `lpApplicationName` 指定要执行的模块；最后两个参数中，`lpStartupInfo` 指向一个 `STARTUPINFO` 结构，`lpProcessInformation` 指向进程信息描述结构。输出进程相关信息的代码如下所示：

```
si.dwFlags = STARTF_USESHOWWINDOW;
si.wShowWindow = TRUE;
BOOL bRet = ::CreateProcess(PROCESS_PATH, NULL, NULL, NULL, FALSE, CREATE_NEW_CONSOLE, NULL, NULL, &si, &pi);

printf("进程句柄: %d\n", pi.hProcess);
printf("线程句柄: %d\n", pi.hThread);
printf("进程的ID号: %d\n", pi.dwProcessId);
printf("进程的主线程ID号: %d\n", pi.dwThreadId);
```

4) 终止进程使用的是 `TerminateProcess()` API 函数，需要根据对象句柄结束进程，相关代码如下所示：

```
hProcess = ::OpenProcess(PROCESS_TERMINATE, FALSE, pe.th32ProcessID); //根据进程ID返回对象句柄
::TerminateProcess(hProcess, 0); //根据对象句柄结束进程
CloseHandle(hSnapshot);
CloseHandle(hProcess);
return;
```

1.6 实验体会（实验中遇到的问题及解决方法）

1) 实验反思

1. 最开始把要执行的模块信息放在 `CreateProcess()` 函数的第二个参数位置，程序运行不出结果，最后发现要把模块信息放在第一个参数位置。

2. 使用不了 `_tcsicmp` 函数，发现需要加上头文件 `#include <tchar.h>`

2) 实验收获

通过本次实验，我对进程的创建和终止有了更深入的理解，学习到了 `CreateProcess()` API 函数和 `TerminateProcess()` API 函数的用法，对于进程运行有了更好的了解。

附录：

《201906021316 马佳敏 OS 实验一源代码》文件夹