



北京理工大学

实验二 进程控制

班 级： 07111505

姓 名： 徐宇恒

学 号： 1120151839

目录

一. 实验目的.....	3
二. 实验内容.....	3
2.1. 在 Windows 下的实现:	3
2.2. 在 Linux 下的实现:	3
三. 实验环境.....	3
四. 实验过程.....	3
4.1. 实验思路:	3
4.1.1. Mytime(主进程)	3
4.1.2. HelloWorld(子进程).....	4
4.2. Windows 实现:	4
4.2.1. 步骤.....	4
4.2.2. 核心代码.....	4
4.3. Linux 实现:	5
4.3.1. 步骤.....	5
4.3.2. 核心代码.....	5
五. 实验结果.....	6
5.1. Windows 运行结果.....	6
5.2. Linux 运行结果	7
六. 心得体会.....	7
七. 附录.....	8

一. 实验目的

分别在 Windows 和 Linux 上实现 Unix 的 time 命令，取名为 mytime

用 mytime 命令记录某可执行程序的运行时间，要求用命令行参数执行该可执行程序，并为该可执行程序创建一个独立的进程。

二. 实验内容

2.1. 在 Windows 下的实现：

- 使用 `CreateProcess()` 来创建进程。
- 使用 `WaitForSingleObject()` 在 “mytime” 命令和新创建的进程之间同步。
- 调用 `GetSystemTime()` 获取时间。

2.2. 在 Linux 下的实现：

- 使用 `fork()/execv()` 来创建进程运行程序。
- 使用 `wait()` 等待新创建的进程结束
- 调用 `gettimeofday()` 来获取时间

三. 实验环境

	Windows	Linux
操作系统	Windows10 Pro 64bit	Ubuntu 16.04LTS
编译器	Visual Studio 2017 IDE	Gcc

四. 实验过程

4.1. 实验思路：

4.1.1. Mytime(主进程)

- 运行 mytime
- 记录开始时间

- 调用子进程
- 阻塞等待子进程结束
- 记录结束时间
- 输出子进程用时

4.1.2. HelloWorld(子进程)

- 输出 “HelloWorld”
- 阻塞等待用户操作

4.2. Windows 实现:

4.2.1. 步骤

- 在 cmd 中运行主进程可执行文件 mytime.exe
- Start = GetSystemTime 获取开始时间
- CreateProcess 创建子进程通过命令行传递参数调用子进程
- WaitForSingleObject 阻塞等待子进程结束
- End = GetSystemTime 获得结束时间
- End - start 计算子程序用时

4.2.2. 核心代码

```
GetSystemTime(&start);  
  
if (!CreateProcess(  
    NULL,  
    argv[1],  
    NULL,  
    NULL,  
    FALSE,
```

```
        CREATE_NEW_CONSOLE,  
        NULL,  
        NULL,  
        &si,  
        &pi)  
    )  
{  
    printf("CreateProcess failed (%d).\n", GetLastError());  
    return 0;  
}  
  
WaitForSingleObject(pi.hProcess, INFINITE);  
  
GetSystemTime(&end);
```

4.3. Linux 实现：

4.3.1. 步骤

- 编译生成可执行文件 mytime
- 运行主进程
- 在主进程中调用 fork() 创建子进程
- 调用 exece() 函数运行可执行程序
- 主进程 wait() 阻塞等待子进程结束
- Gettimeofday() 获取当前时间

4.3.2. 核心代码

```
gettimeofday(&start, NULL);  
  
pid_t pid = fork();  
  
exece(argv[1], arg);
```

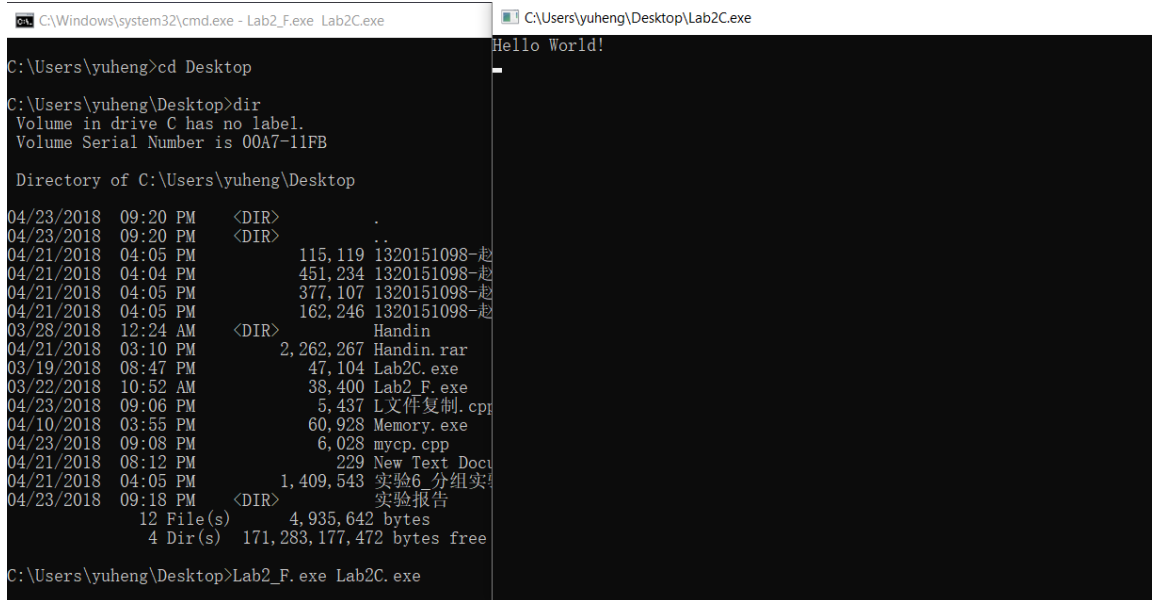
```
wait(NULL);

gettimeofday(&end, NULL);
```

五. 实验结果

5.1. Windows 运行结果

在 cmd 中输入 Lab_F.exe Lab2C.exe 调用子进程成功，等待用户操作



```
C:\Windows\system32\cmd.exe - Lab2_F.exe Lab2C.exe
C:\Users\yuheng>cd Desktop
C:\Users\yuheng\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 00A7-11FB

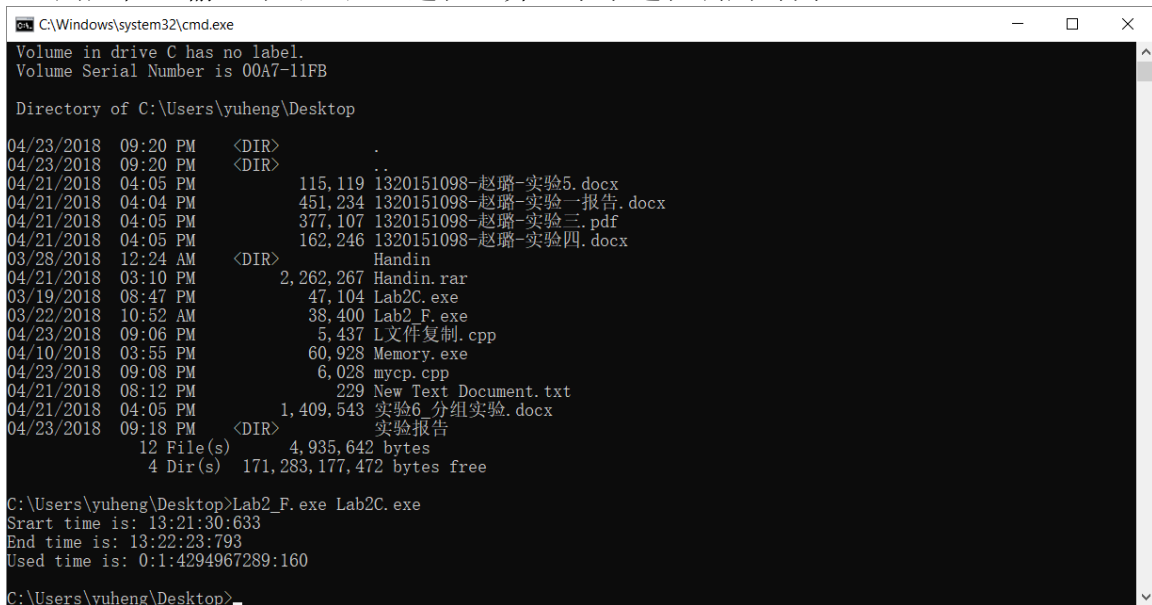
Directory of C:\Users\yuheng\Desktop

04/23/2018  09:20 PM  <DIR>          .
04/23/2018  09:20 PM  <DIR>          ..
04/21/2018  04:05 PM             115,119  1320151098-赵
04/21/2018  04:04 PM             451,234  1320151098-赵
04/21/2018  04:05 PM             377,107  1320151098-赵
04/21/2018  04:05 PM             162,246  1320151098-赵
03/28/2018  12:24 AM  <DIR>          Handin
04/21/2018  03:10 PM      2,262,267  Handin.rar
03/19/2018  08:47 PM      47,104    Lab2C.exe
03/22/2018  10:52 AM      38,400    Lab2_F.exe
04/23/2018  09:06 PM      5,437    L文件复制.cpp
04/10/2018  03:55 PM      60,928    Memory.exe
04/23/2018  09:08 PM      6,028    mycp.cpp
04/21/2018  08:12 PM      229     New Text Docu
04/21/2018  04:05 PM    1,409,543  实验6_分组实
04/23/2018  09:18 PM  <DIR>          实验报告
                           12 File(s)      4,935,642 bytes
                           4 Dir(s)    171,283,177,472 bytes free

C:\Users\yuheng\Desktop>Lab2_F.exe Lab2C.exe

Hello World!
```

用户任意输入即可返回主进程，并显示子进程调用时间



```
C:\Windows\system32\cmd.exe
Volume in drive C has no label.
Volume Serial Number is 00A7-11FB

Directory of C:\Users\yuheng\Desktop

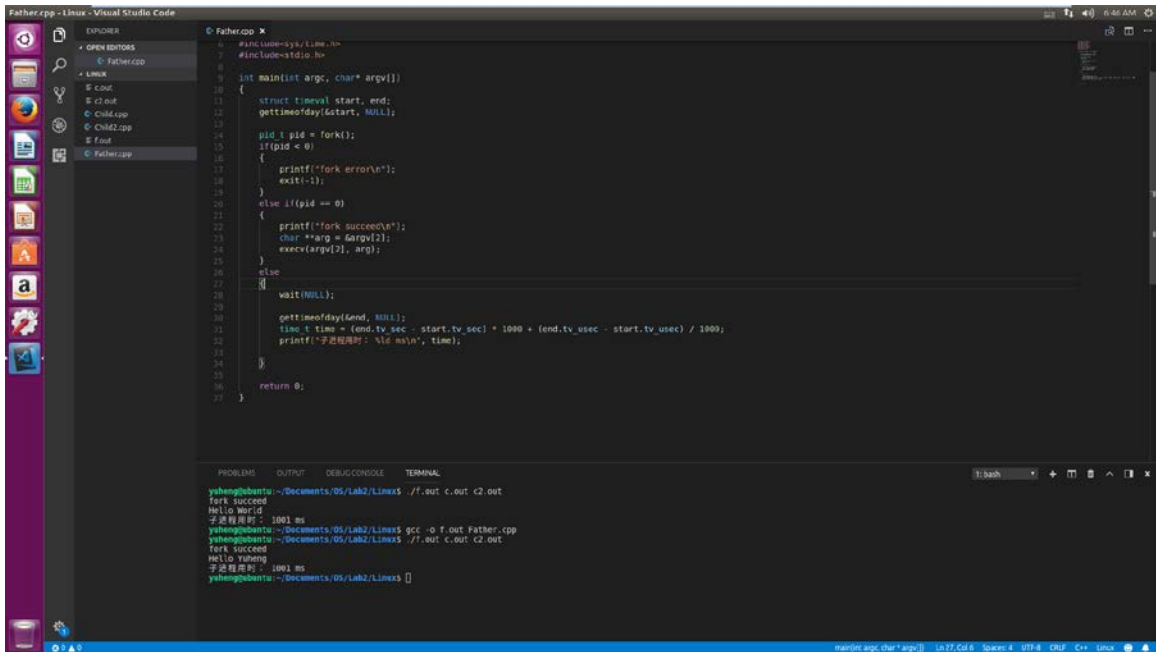
04/23/2018  09:20 PM  <DIR>          .
04/23/2018  09:20 PM  <DIR>          ..
04/21/2018  04:05 PM             115,119  1320151098-赵
04/21/2018  04:04 PM             451,234  1320151098-赵
04/21/2018  04:05 PM             377,107  1320151098-赵
04/21/2018  04:05 PM             162,246  1320151098-赵
03/28/2018  12:24 AM  <DIR>          Handin
04/21/2018  03:10 PM      2,262,267  Handin.rar
03/19/2018  08:47 PM      47,104    Lab2C.exe
03/22/2018  10:52 AM      38,400    Lab2_F.exe
04/23/2018  09:06 PM      5,437    L文件复制.cpp
04/10/2018  03:55 PM      60,928    Memory.exe
04/23/2018  09:08 PM      6,028    mycp.cpp
04/21/2018  08:12 PM      229     New Text Document.txt
04/21/2018  04:05 PM    1,409,543  实验6_分组实验.docx
04/23/2018  09:18 PM  <DIR>          实验报告
                           12 File(s)      4,935,642 bytes
                           4 Dir(s)    171,283,177,472 bytes free

C:\Users\yuheng\Desktop>Lab2_F.exe Lab2C.exe
Start time is: 13:21:30:633
End time is: 13:22:23:793
Used time is: 0:1:4294967289:160

C:\Users\yuheng\Desktop>
```

5.2. Linux 运行结果

通过修改 `argv[1]`和 `argv[2]`，可以调用不同的子程序。如图所示



```
Father.cpp - Linux - Visual Studio Code
1 #include <sys/types.h>
2 #include <stdio.h>
3
4 int main(int argc, char* argv[])
5 {
6     struct timeval start, end;
7     gettimeofday(&start, NULL);
8
9     pid_t pid = fork();
10    if (pid < 0)
11    {
12        printf("fork error\n");
13        exit(-1);
14    }
15    else if (pid == 0)
16    {
17        printf("fork success\n");
18        char **arg = &argv[2];
19        execv(arg[0], arg);
20    }
21    else
22    {
23        wait(NULL);
24
25        gettimeofday(&end, NULL);
26        time_t time = (end.tv_sec - start.tv_sec) * 1000 + (end.tv_usec - start.tv_usec) / 1000;
27        printf("子进程耗时: %ld\n", time);
28    }
29    return 0;
30 }
```

PROBLEMS OUTPUT DEBUGCONSOLE TERMINAL

```
ysheng@buntu:~/Documents/OS/Lab2/Linux$ ./f.out c.out c2.out
fork success
Hello World
子进程耗时: 1001 ms
ysheng@buntu:~/Documents/OS/Lab2/Linux$ gcc -o f.out Father.cpp
ysheng@buntu:~/Documents/OS/Lab2/Linux$ ./f.out c.out c2.out
fork success
Hello ysheng
子进程耗时: 1001 ms
ysheng@buntu:~/Documents/OS/Lab2/Linux$
```

六. 心得体会

通过本次实验，基本了解了 Windows 和 Linux 进程控制的基本方法，明白了如何在主进程中创建创建新的进程，并调用已经写好的可执行文件。

在本次实验中也遇到了很多问题。之前没用接触过 WindowsAPI，对很多参数的含义并不是很明确不过通过阅读 windows 文档以及翻阅教科书逐渐弄明白整个进程的创建过程。

七. 附录

A.1 Windows 源码

A.1.1 mytime.cpp

```
#include<Windows.h>

#include<stdio.h>

#include<stdlib.h>

#include<tchar.h>

#include<time.h>

int main(int argc, char *argv[])
{
    STARTUPINFO si;

    PROCESS_INFORMATION pi;

    SYSTEMTIME start, end;

    ZeroMemory(&memset
    ZeroMemory(&si, sizeof(si));
    ZeroMemory(&pi, sizeof(pi));
    si.cb = sizeof(si);

    if (argc != 2)
    {
        printf("δύ");
        return 0;
    }
}
```



```
}

GetSystemTime(&start);

if (!CreateProcess(
    NULL,
    argv[1],
    NULL,
    NULL,
    FALSE,
    CREATE_NEW_CONSOLE,
    NULL,
    NULL,
    &si,
    &pi)
)
{
    printf("CreateProcess failed (%d).\n", GetLastError());
    return 0;
}

WaitForSingleObject(pi.hProcess, INFINITE);

GetSystemTime(&end);
```

```
        printf("Start time is: %u:%u:%u:%u\n", start.wHour, start.wMinute,
start.wSecond, start.wMilliseconds);

        printf("End time is: %u:%u:%u:%u\n", end.wHour, end.wMinute,
end.wSecond, end.wMilliseconds);

        printf("Used time is: %u:%u:%u:%u\n", (end.wHour - start.wHour),
(end.wMinute - start.wMinute), (end.wSecond - start.wSecond),
(end.wMilliseconds - start.wMilliseconds));

        CloseHandle(pi.hProcess);
    }
```

A.1.2 hellow.cpp

```
#include<cstdio>
#include<iostream>
#include<cstdlib>
#include<Windows.h>
#include<conio.h>
using namespace std;

int main()
{
    printf("Hello World!\n");
    while (1)
    {
        if (kbhit() != 0)
            break;
    }
}
```

```
        return 0;
    }
```

A.2 Linux 源码

A.2.1 mytime.cpp

```
#include<sys/types.h>

#include<unistd.h>

#include<stdlib.h>

#include<sys/wait.h>

#include<sys/types.h>

#include<sys/time.h>

#include<stdio.h>

int main(int argc, char* argv[])
{
    struct timeval start, end;
    gettimeofday(&start, NULL);

    pid_t pid = fork();
    if(pid < 0)
    {
        printf("fork error\n");
        exit(-1);
    }
    else if(pid == 0)
    {
        printf("fork succeed\n");
```

```
        char **arg = &argv[1];
        execv(argv[1], arg);
    }
    else
    {
        wait(NULL);

        gettimeofday(&end, NULL);

        time_t time = (end.tv_sec - start.tv_sec) * 1000 + (end.tv_usec -
start.tv_usec) / 1000;

        printf("子进程用时:  %ld ms\n", time);

    }

    return 0;
}
```

A.2.2 helloWord.cpp

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main()
{
    printf("Hello World\n");
    usleep(1000000);
}
```

```
    return 0;  
}
```

A.2.3 HelloYuheng.cpp

```
#include<stdio.h>  
  
#include<stdlib.h>  
  
#include<unistd.h>  
  
int main()  
{  
    printf("Hello Yuheng\n");  
    usleep(1000000);  
    return 0;  
}
```