

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

第4章 进程管理

-  4.1进程概念
-  4.2进程控制
-  4.3线程
-  4.4临界区和锁
-  4.5同步和P-V操作
-  4.6Windows和Linux同步机制
-  4.7进程通信




华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

4.6 Windows和Linux同步机制

 4.6.1 Windows同步机制

 4.6.2 Linux父子进程同步

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

《操作系统原理》

4.6.2 Linux父子进程同步

教师：苏曙光

华中科技大学软件学院

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

思考：程序运行流程？pid_1和pid_2的值？

```
1  main( )
2  {
3      pid_t pid, pid_1, pid_2 ;
4      pid = fork( );
5      if (pid == 0) { /* 子进程 */
6          pid_1 = getpid( );
7          printf( "pid_1 = %d \n", pid_1 );
8          sleep(10);
9      }
10     if ( pid > 0 ) { /* 父进程 */
11         pid_2 = wait ( NULL );
12         printf( "pid_2 = %d \n", pid_2 );
13     }
14     exit(0);
15 }
```

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

进程的阻塞 wait()

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

■ 进程调用wait(int status)阻塞自己

- 阻塞到有子进程结束？
 - ◆ 没有：进程一直阻塞。
 - ◆ 有（僵尸进程）
 - Wait收集该子进程信息并彻底销毁它后返回。
- status 保存进程退出时的状态。
 - ◆ 若忽略退出信息
`pid = wait(NULL) ;`

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

进程的终结 exit()

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

- 调用void **exit**(int status) 终结进程
- 进程终结时要释放资源并报告父进程
 - 利用status传递进程结束时的状态
 - 变为僵尸状态，保留部分PCB信息供wait收集
 - 正常结束还是异常结束
 - 占用总系统cpu时间
 - 缺页中断次数
 - 调用schedule()函数，选择新进程运行。

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

进程的休眠 sleep()

sleep(int nSecond) .

- 进程暂停执行nSecond秒
- 系统暂停调度该进程
- 相当于windows的suspend()，挂起若干秒。

 华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

思考：程序运行流程？pid_1和pid_2的值？

```
1  main( )
2  {
3      pid_t pid, pid_1, pid_2 ;
4      pid = fork( );
5      if (pid == 0){/* 子进程 */
6          pid_1 = getpid( );
7          printf( "pid_1 = %d \n",pid_1 );
8          sleep(10);
9      }
10     if ( pid > 0 ){ /* 父进程 */
11         pid_2 = wait ( NULL );
12         printf( "pid 2 = %d \n",pid_2 );
13     }
14     exit(0);
15 }
```

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

父子进程共享普通变量

```
int main(int argc, char *argv[ ])
{
    pid_t pid;
    int i = 1;
    pid = fork( ); //创建新进程
    if(pid == 0) //子进程
    {
        i = 2;
        printf( "In child i = %d\n", i ); //打印 i 值
        exit( 0 );
    }
    else //父进程
    {
        sleep(10); //10秒休眠，让子进程先执行
        printf( "In parent i = %d\n", i ); //打印 i 值
        exit( 0 );
    }
}
```

思考：父子进程输出i是多少？

[root@michael root]# ./test

In child i = 2

In parent i = 1

结论：对于普通变量，父子进程各自操作变量副本，互相不影响。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

父子进程共享文件资源

```
int main(int argc, char *argv[ ])
{
    int file;
    char *chA = "A" ;
    char *chB = "B" ;
    char *chC = "C" ;
    //打开（创建）一个文件
    file = open( "test.txt" );
    write( file, chA, strlen(chA));
    pid_t pid = fork( ); //创建新进程
    if(pid == 0) //子进程
    {
        write( file, chB, strlen(chB)); //写test.txt
        exit( 0 );
    }
    else //父进程
    {
        sleep( 10 ); //休眠10秒，让子进程先运行完
        write( file, chC, strlen(chC)); //写test.txt
        exit( 0 );
    }
}
```

思考：程序运行后test.txt内容是什么？

test.txt内容是：

ABC或ACB或A或B或C或AC或AB或.....?

结论：对于文件，父子进程
共享同一文件和读写指针。

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有