

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

4.2进程控制

-  4.2.1进程控制概念
-  4.2.2Windows进程控制
-  4.2.3Linux进程控制

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》

4.2.3 Linux进程控制



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

创建进程——fork()

```
pid_t fork(void);
```

例：`pid_t pid = fork();`

- ◆ 新进程是当前进程的子进程。
- ◆ 父进程和子进程
 - ◆ 父进程：fork()的调用者
 - ◆ 子进程：新建的进程
- ◆ 子进程是父进程的复制。
- ◆ 父进程和子进程并发运行。

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

思考：下面程序在屏幕上将输出什么内容？



// 文件名 testfork.c

```
int main(void)
{
    fork( );
    printf( "Hello World!\n" );
    return 0;
}
```

子进程是父进程的复制。

父进程和子进程并发运行。

```
[root@localhost ~]# ./testfork
Hello World!
Hello World!
```


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

思考：下面程序运行结果是什么？

□ fork() 函数创建进程

```
int main(void)
{
    pid_t pid;
    pid = fork();
    if( pid == 0 )
        printf( "Hello World!\n" );
    else if( pid > 0 )
        printf( "How are you!\n" );
}
```

```
[root@host ~]# ./fork
Hello World!
How are you!
```

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

创建进程——fork()

```
pid_t fork(void);
```

例：`pid_t pid = fork();`

- fork 返回值pid

- ◆在子进程中，`pid = 0`

- ◆在父进程中，`pid > 0`（子进程ID）

- ◆出错：`pid = -1`

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

思考：下面程序运行结果是什么？

□ **fork()函数创建进程**

```
int main(void)
```

```
{
```

```
    pid_t pid;
```

```
    pid = fork( );
```

```
    if( pid == 0 )
```

```
        printf( "Hello World!\n" );
```

```
    else if(pid > 0)
```

```
        printf( "How are you!\n" );
```

```
}
```

```
[root@host ~]# ./fork
```

```
Hello World! ✓
```

```
How are you! ✓
```


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

思考：下面程序运行结果是什么？

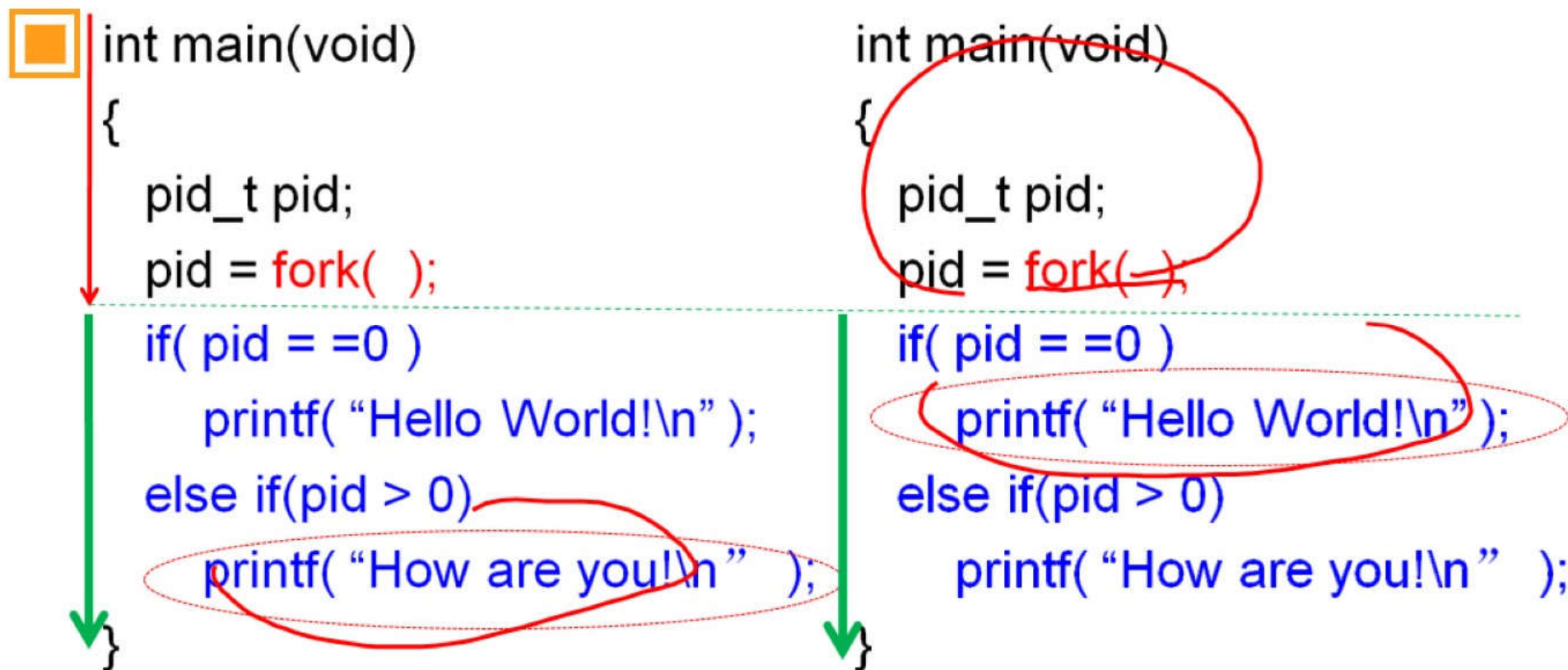
□ **fork()函数创建进程**

```
int main(void)
{
    pid_t pid;
    pid = fork( );
    if( pid == 0 )
        printf( "Hello World!\n" );
    else if(pid > 0)
        printf( "How are you!\n" );
}
```

```
[root@host ~]# ./fork
How are you! ✓
Hello World! ✓
```

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

fork()执行的流程



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

fork函数的实现

//在文件/kernel/fork.c中

int do fork(

unsigned long clone_flags,
unsigned long stack_start,
struct pt_regs *regs,
unsigned long stack_size

);

- 为子进程分配task_struct空间；
- 初始化子进程task_struct；
- 复制父进程的file，fs，sighand，mm等信息；
-

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

fork函数的实现

```
int do_fork {  
    struct task_struct *p; ✓  
    //分配物理页面存放task_struct结构和内核空间的堆栈  
    p = alloc_task_struct( ); ✓  
    //把当前进程task_struct结构中所有内容都拷贝到新进程中  
    *p = *current;  
    //判断用户进程数量是否超过了最大限制，否则不许fork  
    if ( (&p->user->processes) >= p->rlim[RLIMIT_NPROC].rlim_cur))  
        goto bad_fork_free;  
    //子进程初始状态设TASK_UNINTERRUPTIBLE  
    p->state = TASK_UNINTERRUPTIBLE;
```


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

fork函数的实现

```
// 拷贝进程的所有信息
copy_files(clone_flags, p);
copy_fs(clone_flags, p);
copy_mm(clone_flags, p);
//进程创建后与父进程链接起来形成一个进程组
list_add(&p->thread_group, &current->thread_group);
//唤醒进程，将其挂入可执行队列等待被调度
wake_up_process(p); ✓

} // do_fork 结束
```

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

进程执行与父进程不同的功能

exec函数簇（包括若干函数）

■功能

- ◆装入一个指定的可执行程序运行
- ◆使子进程具有和父进程完全不同的新功能。

■步骤

- ◆根据文件名找到相应的可执行程序。
- ◆将可执行程序的内容填入子进程的地址空间
- ◆进入新进程执行且不再返回。

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

exec函数族



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

```
int execl ( const char *path, const char *arg, ...);
```

```
int execlp ( const char *file, const char *arg, ...);
```

```
int execl_e ( const char *path, char *arg, ..., char *envp[ ]);
```

```
int execl_v ( const char *path, char *const argv[]);
```

```
int execl_vp (const char *file, char *const argv[]);
```

```
int execve (const char *path, char *const argv[], char *const envp[]);
```