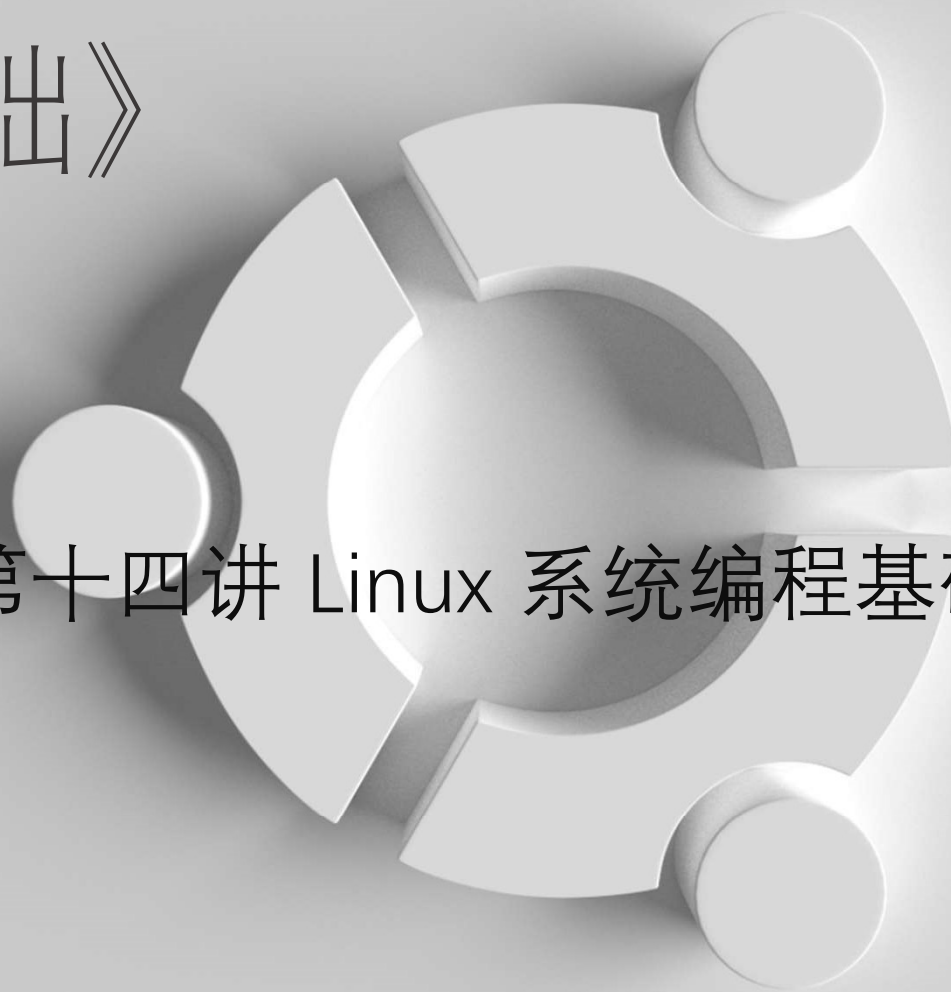


《Linux基础》

第十四讲 Linux 系统编程基础



系统编程简介

- 系统编程就是调用Linux系统提供的API完成需要的任务。
- Linux上的大多数命令都是用C编写的，多数都需要用到系统调用。
- `man 2 [system call name]`查看系统接口文档，`man 3 [lib function]` 查看程序库函数的文档。文档开头都会说明需要引入的头文件，函数声明等信息。
- `man syscalls`查看所有系统调用（API）。
- 本次课程讲解基本的系统调用，主要包括获取进程ID，fork创建子进程，open，write，close操作文件，IO重定向如何实现等内容。

获取自己的PID

- 系统调用: `pid_t getpid();`

- 示例:

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

```
int main(int argc, char *argv[]) {
    printf("%d\n", getpid());
    return 0;
}
```

用fork创建子进程

- 系统调用: `pid_t fork();`
- `fork`会创建子进程, 调用`fork`, 新创建的进程会和父进程一样继续执行。
- `fork`出错返回-1并且不会创建新的进程; 正确则在父进程返回创建子进程的PID, 在子进程返回0。
- 由于父进程和子进程不同的返回值。可以通过判断返回值控制父进程和子进程执行不同的代码。

等待子进程退出

- 系统调用: `pid_t wait(int *status);`
- `wait`等待子进程退出, 并把子进程退出状态设置到`status`变量。返回退出进程的PID。
- `wait`调用会挂起父进程, 直到子进程退出。
- 类似的调用还有`pid_t waitpid(pid_t pid, int *status, int options);`详细说明可在终端运行 `man 2 waitpid`查看。
- `wait`调用相当于调用`waitpid(-1, &status, 0);`

父进程先于子进程退出

- 父进程退出后，子进程继续执行，此时谁又是子进程的父进程？

父进程先于子进程退出，子进程被init进程接管

- 父进程退出后，子进程继续执行，此时父进程是init（ID为1的进程）。而在终端运行程序，当前shell是父进程的父进程，但是由于父进程的提前退出，导致子进程被init进程接管。
- 这是Linux的设计方式，并且此方式是实现守护进程的基础。

open函数

- open函数用于打开文件操作:

int open(const char *pathname, int flags, mode_t mode); 参数依次为文件路径名称, 标志位, 模式。成功返回值为打开的文件描述符, 错误返回-1。

flags选项:

O_CREAT	没有则创建文件
O_WRONLY	写模式打开文件
O_RDONLY	只读模式打开文件
O_RDWR	读写方式打开

mode选项:

S_IRWXU	文件所有者具有可读, 可写, 可执行的权限
S_IRUSR	文件所有者有可读权限
S_IWUSR	文件所有者具有可写权限

write函数

- write函数向一个文件写入数据：
 `ssize_t write(int fd, const void *buf, int count);`
 参数依次为打开的文件描述符，指向数据的指针，要写入的字节数。
 返回值是成功写入的字节数，错误则返回-1。
- 在文件操作最后要记得使用close关闭打开的文件：`int close(int fd);` close函数成功返回0，错误返回-1。

IO重定向

- IO重定向基于这样一个设计原则：最低可用文件描述符（Lowest Available fd）原则。
- 文件描述符是一个数组索引号，每个进程都有一组打开的文件，这些打开的文件信息保存在一个数组中，文件描述符就是数组的索引号。
- 在打开文件时，分配的描述符总是数组中最低可用的索引位置（索引数字最小的位置）。
- 在Linux上，使用0，1，2作为程序的标准输入，标准输出，标准错误输出。而如果关闭描述符1，然后打开其他文件，这样文件就被分配了文件描述符1，于是标准输出就会写入到新打开的文件。这就是IO重定向。

IO重定向实现方式

- 编程实现的方式（以文件描述符1为例）：
 - close-open-close方式，先close(1)，然后open(filename, O_RDWR, S_IWUSR)；操作完成，close关闭新打开的文件。
 - open-close-dup-close方式，open打开文件，返回的文件描述符不是1，然后close(1)，现在最低可用描述符是1，dup(fd)会把新打开的描述符复制到1，然后close(fd)关闭新打开的文件。
 - open-dup2-close方式，dup2(oldfd, newfd)，关闭newfd，把oldfd复制到newfd，close关闭新打开的描述符。