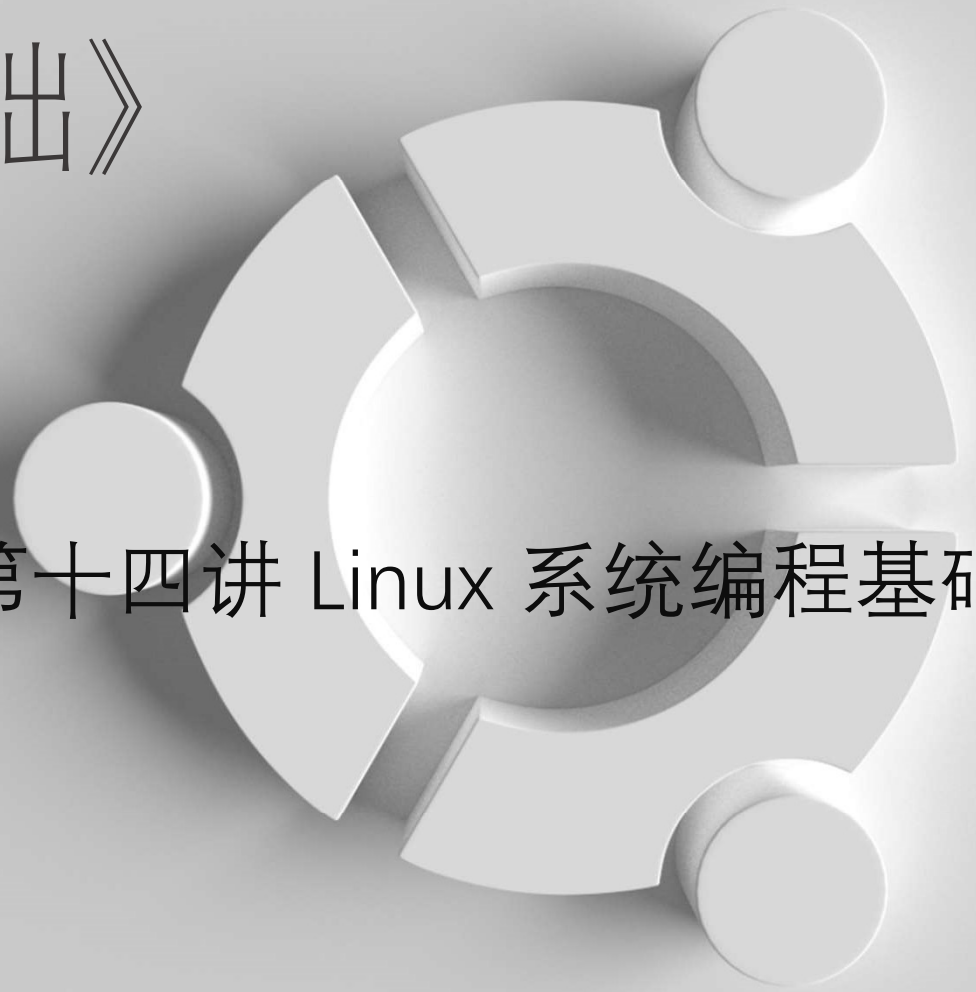
河北师范大学 软件学院
Software College of Hebei Normal University

《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。