操作系统

Operating system

吴国伟

大连理工大学

内容纲要

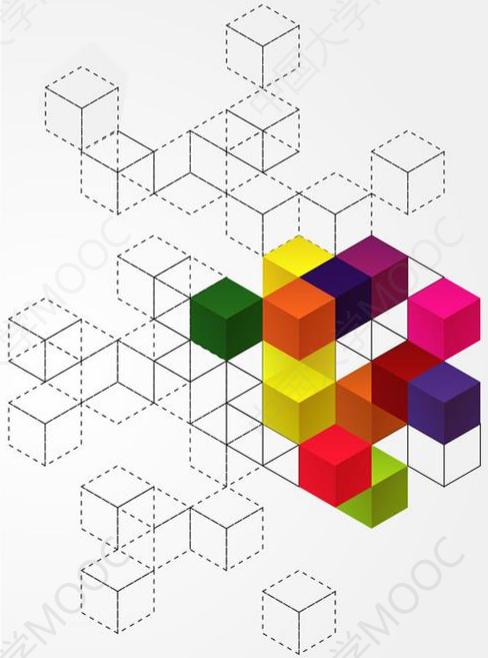
2.4 Linux系统调用实现

一、 Linux系统调用概述

二、 Linux系统调用流程

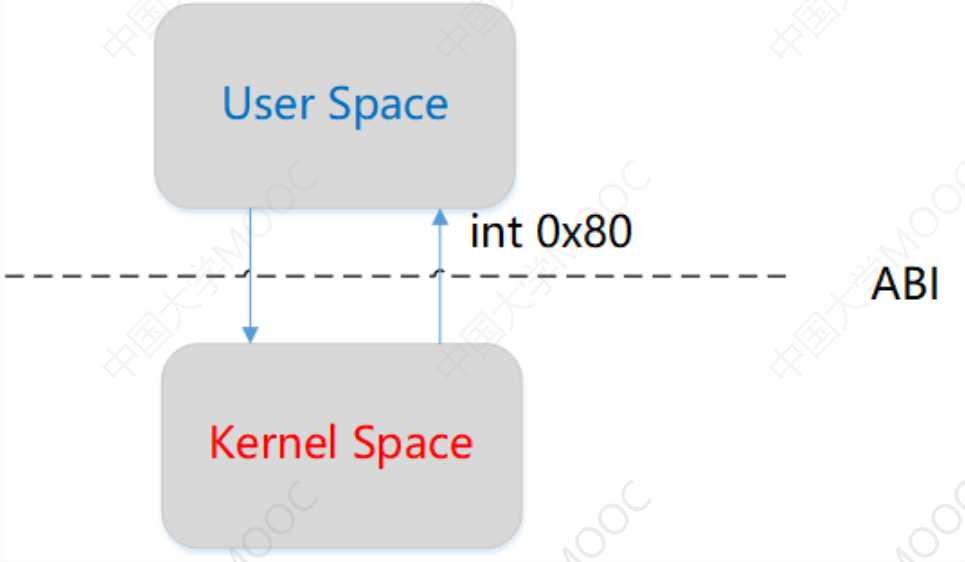
三、 系统调用参数传递

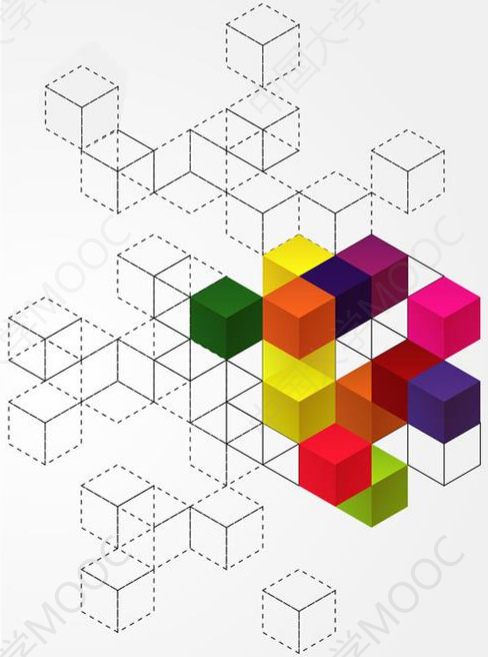
四、 典型Linux系统调用



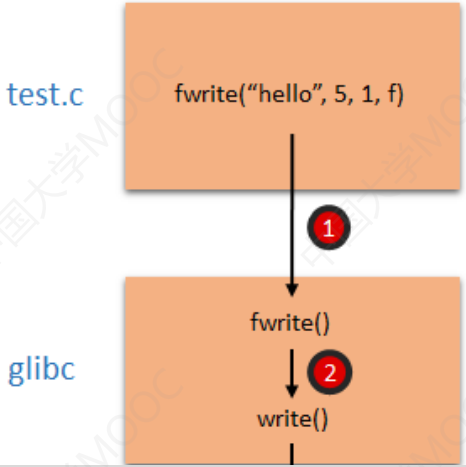
一、Linux系统调用概述

- The system call is the fundamental interface between an application and the Linux kernel.

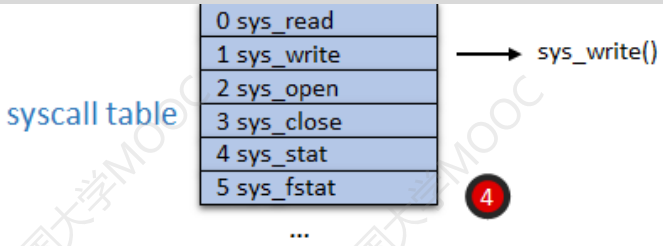


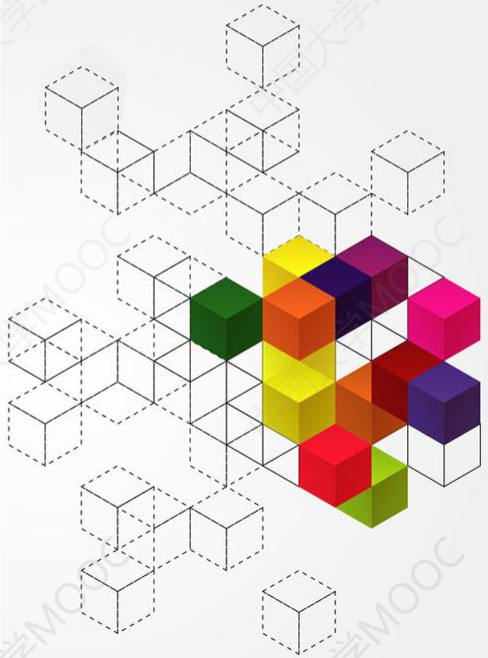


一、Linux系统调用概述



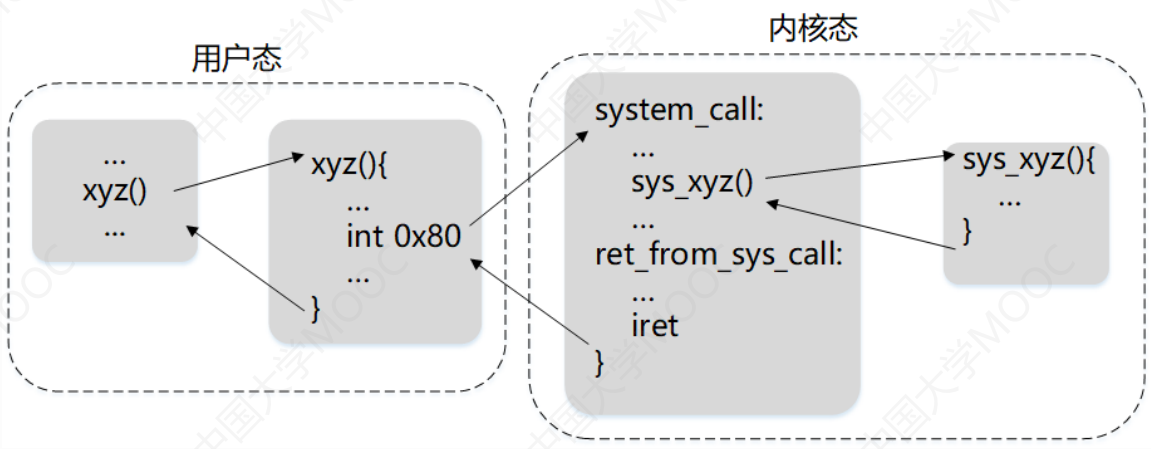
- System calls are generally invoked via wrapper functions in glibc

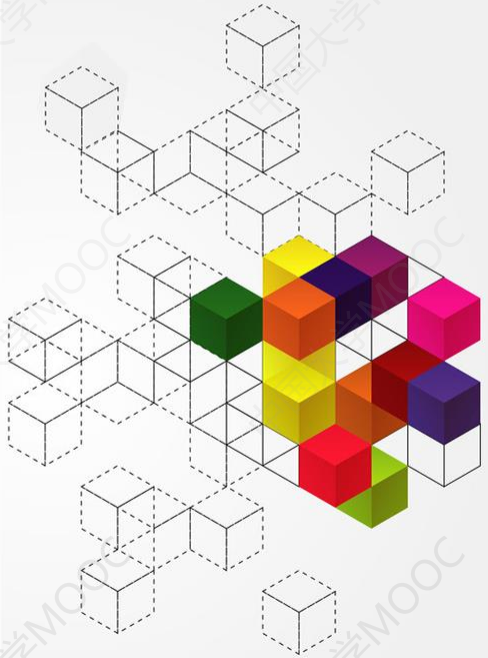




二、Linux系统调用流程

- **Linux系统调用实际执行流程示意图**





二、Linux系统调用流程

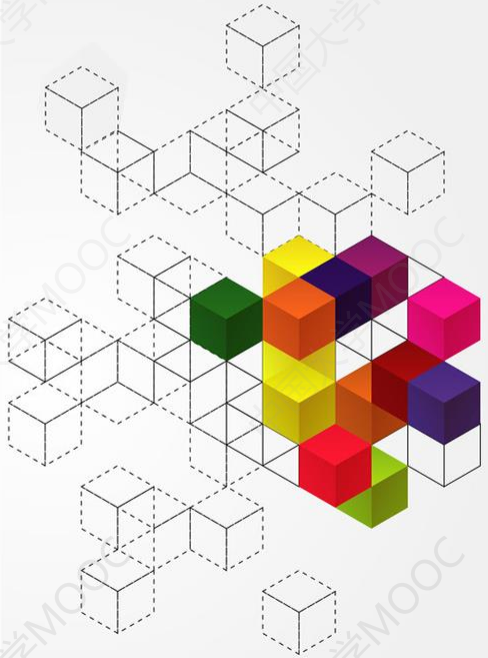
- **Linux系统调用实现：基于软中断**
  - int 0x80

The Linux kernel registers an interrupt handler named ia32_syscall for the interrupt number: 128 (0x80).

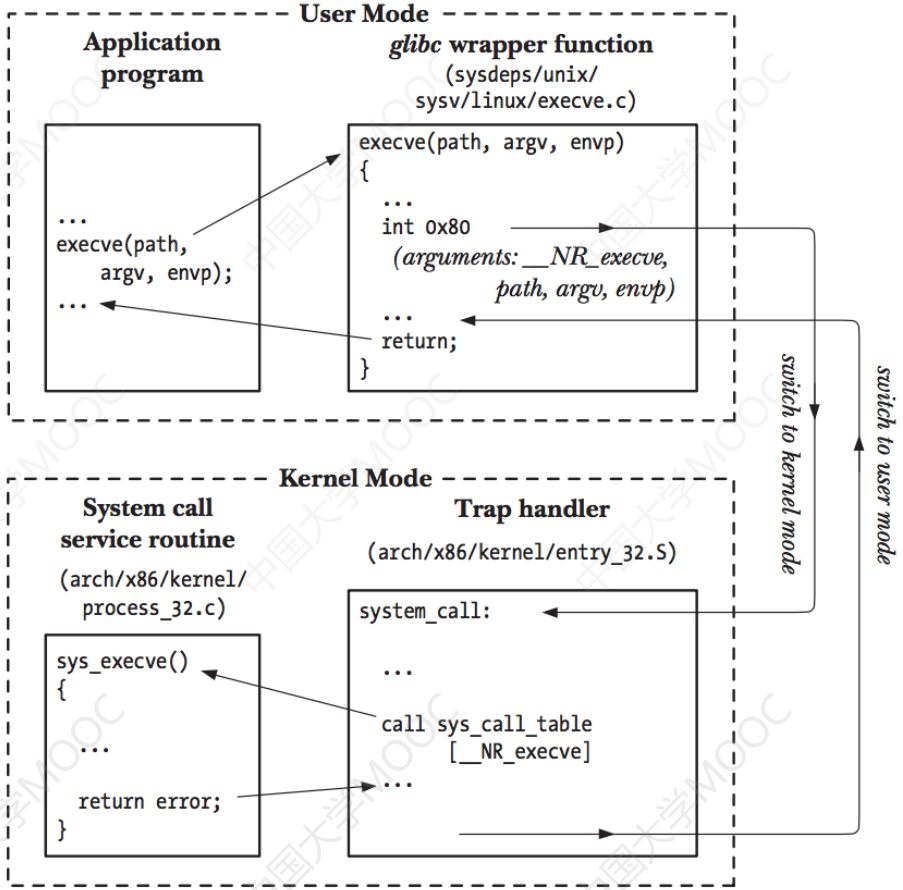
From the `trap_init` function in the kernel 3.13.0 source in `arch/x86/kernel/traps.c` :

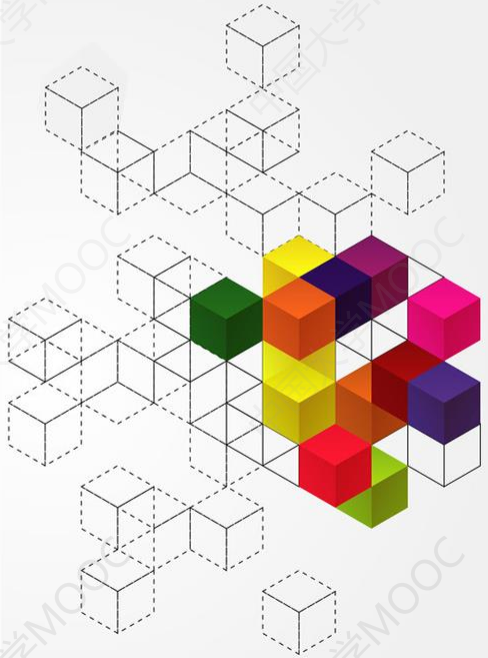
```
void __init trap_init(void)
{
        /* ..... other code ... */

        set_system_intr_gate(IA32_SYSCALL_VECTOR, ia32_syscall);
```

Where `IA32_SYSCALL_VECTOR` is a defined as `0x80` in `arch/x86/include/asm/irq_vectors.h` .



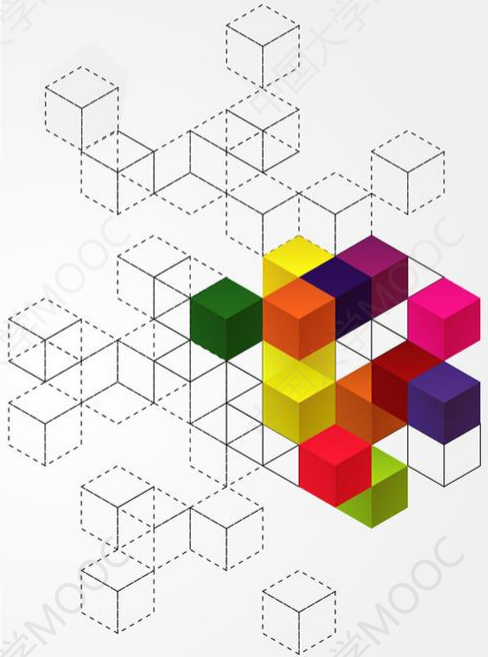
二、Linux系统调用流程





三、系统调用参数传递

- **传递参数的类型**
  - 值传递
  - 传递变量的地址
  - 传递函数指针（例如，信号处理函数指针）



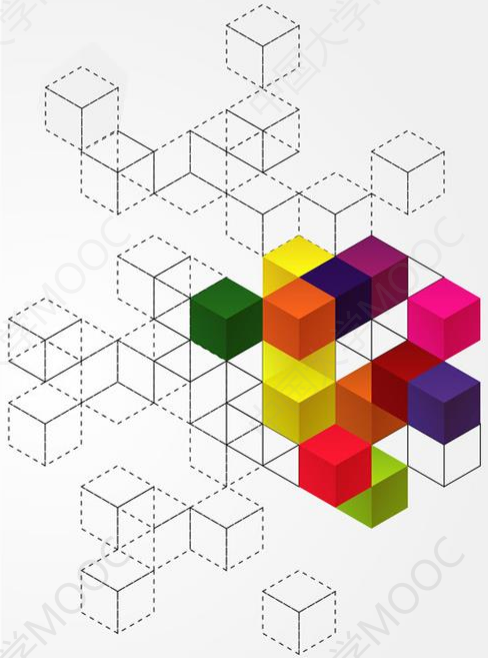
三、系统调用参数传递

寄存器传参（ia32示例）

```
379     * Emulated IA32 system calls via int 0x80.
380     *
381     * Arguments:
382     * %eax System call number.
383     * %ebx Arg1
384     * %ecx Arg2
385     * %edx Arg3
386     * %esi Arg4
387     * %edi Arg5
388     * %ebp Arg6    [note: not saved in the stack
```

from arch/x86/ia32/ia32entry.S, linux v3.13

第1个参数：系统调用号（eax）
寄存器传参最多支持6个参数



四、典型Linux系统调用

64-bit | 32-bit (Coming soon)

Instruction: `syscall`

Return value found in: `%rax`

Syscalls are implemented in functions named as in the *Entry point* column, or with the `DEFINE_SYSCALLx(%name%)` macro.

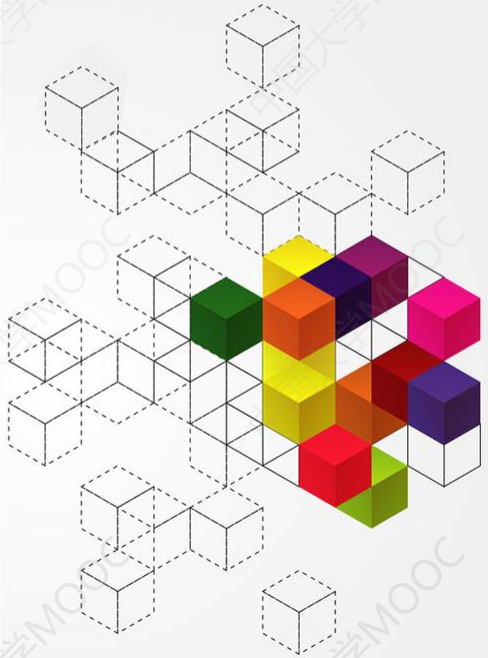
Relevant man pages: `syscall(2)`, `syscalls(2)`

Double click on a row to reveal the arguments list. Search using the fuzzy filter box.

Filter:

%rax	Name	Entry point	Implementation
0	read	sys_read	fs/read_write.c
1	write	sys_write	fs/read_write.c
2	open	sys_open	fs/open.c
3	close	sys_close	fs/open.c
4	stat	sys_newstat	fs/stat.c
5	fstat	sys_newfstat	fs/stat.c
6	lstat	sys_newlstat	fs/stat.c

Linux系统调用表（快速检索）



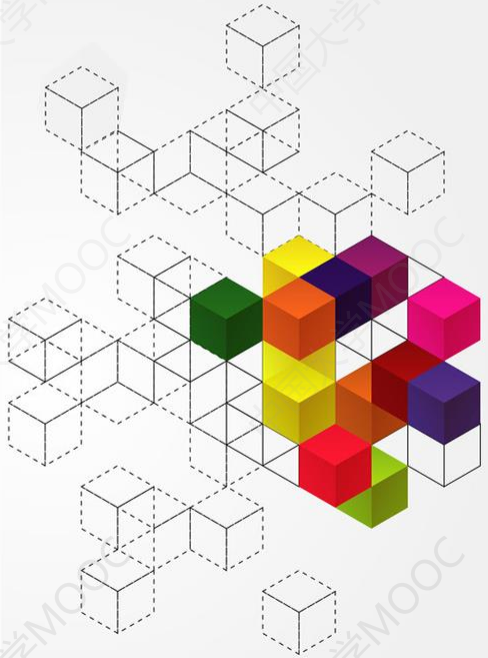
300多个系统调用可以轻松查阅

https://filippo.io/linux-syscall-table/

四、典型Linux系统调用

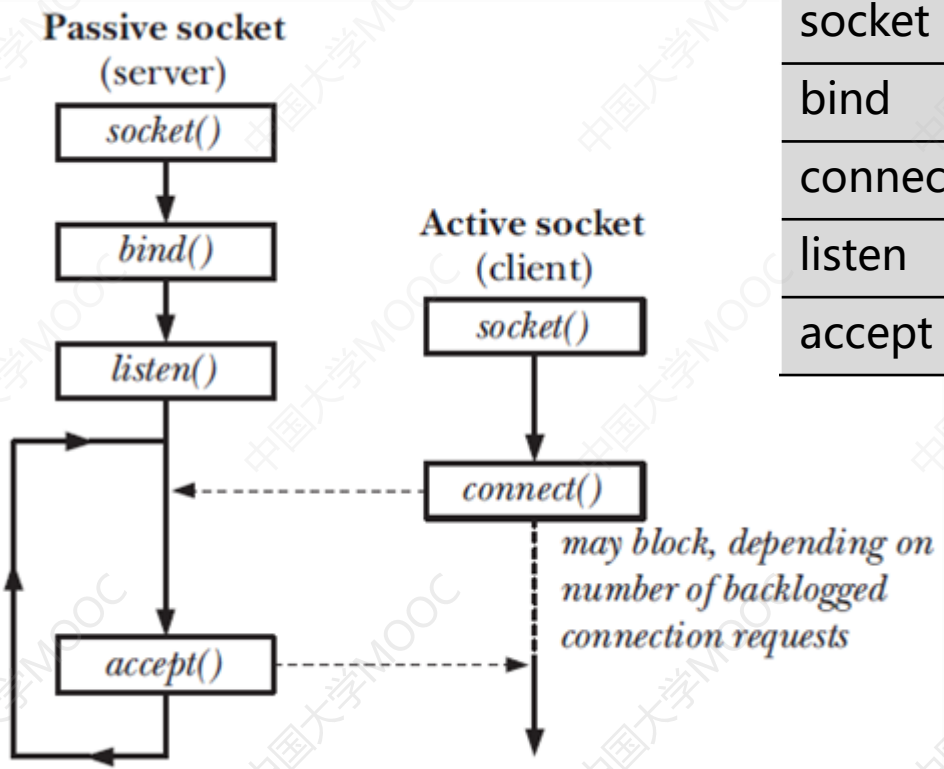
- 常用文件系统调用

名称	功能	函数原型
open	打开	int open(const char *pathname,int flags) int open(const char *pathname,int flags, mode_t mode)
creat	创建	int creat(const char *pathname, mode_t mode)
close	关闭	int close(fd)
lseek	定位	off_t lssek(int fd, off_t offset, int whence)
read	读取	ssize_t read(int fd, void *buf, size_t count)
write	写入	ssize_t write(int fd, const void *buf, size_t count)

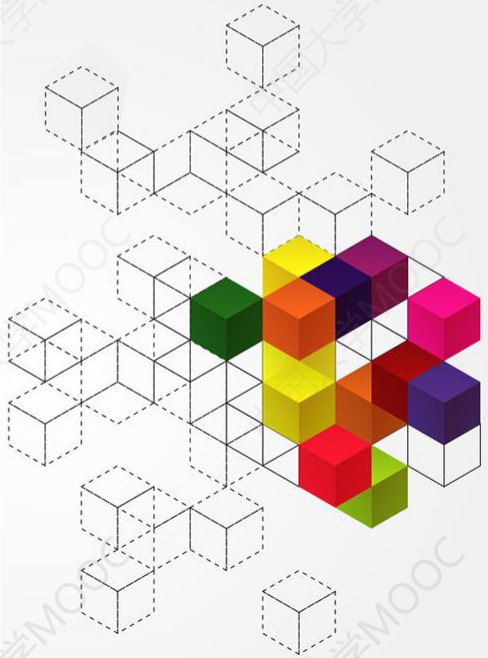


四、典型Linux系统调用

- 网络通信



名称	功能
socket	创建套接字
bind	绑定端口
connect	发起连接
listen	监听
accept	接受连接



本讲小结

- Linux系统调用概述
- Linux系统调用流程
- Linux系统调用参数传递
- 典型Linux系统调用