

Lesson 14

1.read 函数的原型为：`ssize_t read(int fd, void *buf, size_t count);`，它的返回值有以下情形：

- a) 成功返回读取的字节数
- b) 遇到 EOF 返回 0
- c) 读取失败返回-1，并设置相应的 `errno`

2.如果调用 `read(fd, buf, 100)`去读取长度为 230 的文件：

- a) 第一次返回 100
- b) 第二次返回 100
- c) 第三次返回 30
- d) 第四次返回 0
- e) **read 函数调用 4 次。**

3.`readn` 保证在没有遇到 EOF 的情况下，一定可以读取 `n` 个字节。它的返回值有三种：

- a) `>0`，表示成功读取的字节数，如果小于 `n`，说明中间遇到了 EOF
- b) `==0` 表示一开始读取就遇到 EOF
- c) `-1` 表示错误（**这里的 `errno` 绝对不是 `EINTR`。**）。

4.`writen` 函数保证一定写满 `n` 个字节，返回值：

- a) `n` 表示写入成功 `n` 个字节
- b) `-1` 写入失败（这里也没有 `EINTR` 错误）

5.readline 函数： `ssize_t readline(int fd, void *usrbuf, size_t maxlen)`，它的返回值：

- a) 错误返回-1，不包括 EINTR
- b) 读取过程中碰到\n
- c) 没有碰到\n，而是读满 maxlen-1 个字节

6.readn、writen、readline 属于同一个系列，称为网络编程三大函数。

7.RIO 中 rio_read 函数的编写思想：

- a) 采用预读取方案，提前把数据读入 Buffer。
- b) 每当用户取数据的时候，从 Buffer 里面读取，而不是使用系统调用 read 函数，这样减少了多次使用系统调用的开销。

8.rio_read 函数的编写原则：必须与系统的 read 函数保持语义一致，这意味着 rio_read 的返回值有三种情况：

- a) -1 代表出错，这里不包含 EINTR
- b) 0 代表 EOF，读取结束
- c) >0 表示读取的字节数

9.操作系统有三大抽象概念：

- a) 进程：程序的执行过程
- b) 文件：IO
- c) 虚拟内存：可用的地址空间

10.进程在内核中是一种数据结构 task_struct。

11.进程由 PCB、代码段以及数据段组成，其中 **PCB 位于内核空间**。

12.虚拟内存的前面一部分为内核空间，其他的属于用户空间。

13.进程的运行模式有用户态和内核态，以 read 为例，如果接收 1000 个字节，那么需要进行系统调用，**由内核去接收 1000 个字节**，然后在返回用户态运行 read 的时候，**把数据拷贝到用户空间**。

14.用户态到内核态有两种方式，一是**系统调用**，触发 trap 指令陷入内核，二是**中断**，二者的区别在于前者是**自愿**的，后者是**被动**的。

15.进程由三种经典状态

- a) 就绪：准备完毕，随时等待调度
- b) 运行：占有 CPU
- c) 阻塞：等待某一事件的发生

16.三种状态之间的转化：

- a) 就绪到运行 被调度
- b) 运行到阻塞：**执行了 IO 等需要等待的系统调用**，例如 read
- c) 阻塞到就绪：等待的事件来临，**例如 read 所等待的 fd 中有数据可读**。
- d) 运行到就绪：时间片到期或者被抢占

17.不存在阻塞到运行：因为调度器总是从就绪队列里面挑选进程。

18.僵尸进程：子进程退出，而父进程没有对其进行回收。这里注意：

- a) 僵尸进程占用的不是用户空间的资源，子进程运行过程中申请的资源已经全部被回收。
- b) 占用的是内核中的某些结构，例如 PCB，它主要是留给父进程做参考，以便于父进程获取子进程的运行状态。

19.孤儿进程：父进程先退出，子进程托管给 init 进程。

20.利用 fork 产生的父子进程，地址空间是独立的。因为在传统的 UNIX 进程模型中，fork 将父进程的地址空间复制了一份。

21.vfork 在产生子进程的时候，没有复制地址空间，而是与父进程共享。vfork 的目的就是为了在子进程中实行 exec 替换。

22.atexit 作用是向系统注册一些函数，这些函数在进程退出的时候被调用。这些函数被调用的顺序与他们被注册的顺序相反。

23.exit 与 _exit 的区别：

- a) exit 会清空 IO 缓冲区，后者不会
- b) exit 会处理通过 atexit 注册的函数