

1. 由open和openat函数返回的文件描述符一定是最小的未用的描述符数值。

2. 函数: P50-53

`#include <fcntl.h>`

1). `open(const char *path, int oflag, ... /*mode_t mode */);`

功能: 打开或创建一个文件;

返回:

成功: 文件描述符;

失败: -1

2). `openat(int fd, const char *path, int oflag, ... /* mode_t mode */);`

功能: 打开或创建一个文件;

返回:

成功: 文件描述符;

失败: -1

3). `int creat(const char *path, mode_t mode);`

功能: 创建一个新文件, 并以只写方式打开; //不常用了

返回:

成功: 只写打开的文件描述符;

失败: -1

`#include <unistd.h>`

4). `int close (int fd);`

功能: 关闭一个打开的文件;

返回:

成功: 0;

失败: -1

5). `off_t lseek(int fd, off_t offset, int whence);`

功能: 显式地为一个打开文件设置偏移量;

返回:

成功: 新的文件偏移量;

失败: -1

参数:

whence:

SEEK\_SET: 将文件的偏移量设置为 距文件开始处 offset个字节;

SEEK\_CUR: ..... 其当前值加offset, offset可正可负;

SEEK\_END: ..... 文件长度加offset, offset可正可负;

6). `ssize_t read(int fd, void *buf, size_t nbytes);`

功能: 从打开的文件中读数据;

返回:

成功: 读到的字节数, 若已到文件尾, 返回0;

失败: 若出错, -1;

7). `ssize_t write(int fd, const void *buf, size_t nbytes);`

功能: 向打开的文件写数据;

返回:

成功: 返回已经写的字节数;

失败: -1;

8). `ssize_t pread(int fd, void *buf, size_t nbytes, off_t offset);`

功能: 相当于调用lseek()后调用read();(属于原子操作)

返回:

成功: 读到的字节数, 若已到文件尾, 返回0;

失败: -1;

9). `ssize_t pwrite(int fd, const void *buf, size_t nbytes, off_t offset);`

功能: 相当于调用lseek()后调用write();(属于原子操作);

返回:

成功: 返回已经写的字节数;

失败: -1;

10). `int dup(int fd);`

功能: 复制一个现有的文件描述符

返回:

成功：新的文件描述符；  
失败：-1；

11). int dup2(int fd, int fd2);

功能：复制一个现有的文件描述符；

返回：

成功：新的文件描述符；  
失败：-1；

12). int fsync(int fd);

功能：-----;

返回：

成功：0；  
失败：-1；

13). int fdatasync(int fd);

功能：-----;

返回：

成功：0；  
失败：-1；

14). void sync(void);

功能：-----;

返回：

成功：无；  
失败：无；

//include <fcntl.h>

15). int fcntl(int fd, int cmd, .../\* int arg \*/);

功能：改变已经打开文件的属性;(五种功能：p65)

返回：

成功：依赖于cmd；  
失败：-1；

//include <unistd.h>

16). int ioctl(int fd, int request, ...);

功能：I/O操作的杂物箱...;

返回：

成功：其他值；  
失败：-1；

3. 当一个进程终止时，内核会自动关闭它所有打开的文件。

4. 每一个打开的文件都有一个与其关联的"当前文件偏移量"(current file offset);