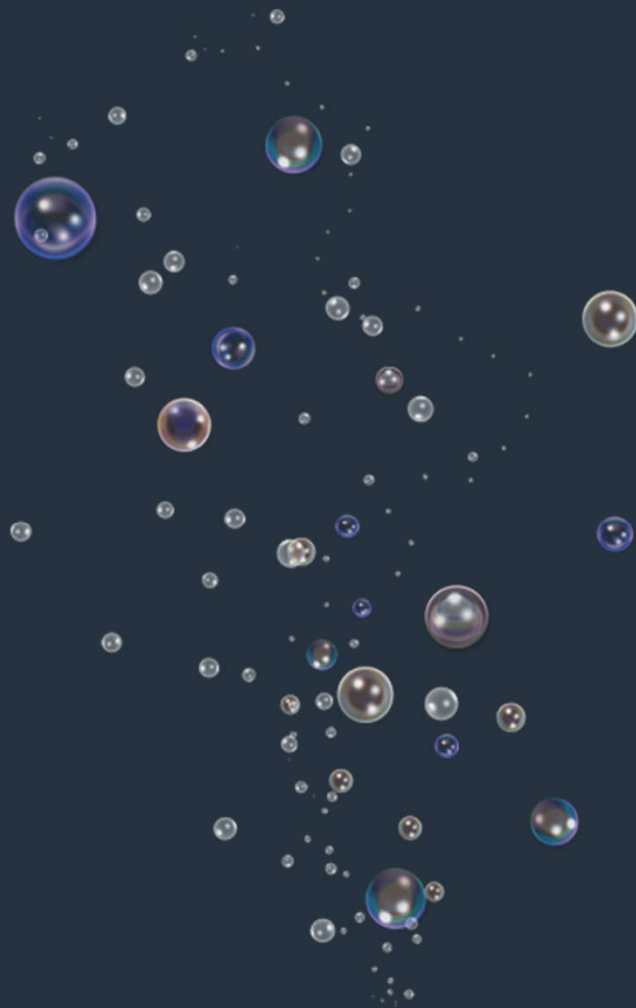




# 内存映射

大连理工大学 赖晓晨



## 内存映射文件

- ✓ Linux允许任何进程把一个磁盘文件映射到内存中，在磁盘文件和它在内存中的映射间创建逐字节对应的关系。
- ✓ 内存映射文件的优点
  - 相对于到磁盘上存取文件来说速度快
  - 可以在内存中共享数据
- ✓ Linux提供一系列调用管理内存映射，应用于framebuffer等场合。

## mmap函数

- ✓ mmap函数用来建立一个磁盘文件的内存映射，返回指向此内存首地址的指针。

```
#include <unistd.h>
#include <sys/mman.h>
void *mmap( void *start, size_t length,
            int prot, int flags, int fd, off_t offset );
```

## mmap函数的参数

- ✓ start: 指向欲对应的内存起始地址, 通常设为NULL, 代表让系统自动选定地址, 对应成功后该地址会返回。

```
#include <unistd.h>
#include <sys/mman.h>
void *mmap( void *start, size_t length,
            int prot, int flags, int fd, off_t offset    );
```

## mmap函数的参数 (续)

- ✓ length: 将文件中多大的部分映射到内存

```
#include <unistd.h>
#include <sys/mman.h>
void *mmap( void *start, size_t length,
            int prot, int flags, int fd, off_t offset );
```

## mmap函数的参数 (续)



prot: 映射区域的保护方式, 有下列组合:

- PROT\_EXEC 映射区域可被执行
- PROT\_READ 映射区域可被读取
- PROT\_WRITE 映射区域可被写入
- PROT\_NONE 映射区域不能访问

```
#include <unistd.h>
#include <sys/mman.h>
void *mmap( void *start, size_t length,
            int prot, int flags, int fd, off_t offset );
```

## mmap函数的参数 (续)



flags: 影响映射区域的各种特性:

- MAP\_SHARED: 对映射区域的写入数据会复制回文件, 且允许其他映射该文件的进程共享。
- MAP\_PRIVATE: 对映射区域的写入操作会产生一个映射文件的复制, 对此区域作的任何修改都不会写回原来的文件内容。

```
#include <unistd.h>
#include <sys/mman.h>
void *mmap( void *start, size_t length,
            int prot, int flags, int fd, off_t offset    );
```

## mmap函数的参数 (续)

- ✓ fd: 为open()返回的文件描述符, 代表欲映射到内存的文件。

```
#include <unistd.h>
#include <sys/mman.h>
void *mmap( void *start, size_t length,
            int prot, int flags, int fd, off_t offset    );
```



## mmap函数的参数 (续)

- ✓ offset: 参数offset为文件映射的偏移量, 通常设置为0, 代表从文件最前方开始对应, offset必须是分页大小的整数倍。

```
#include <unistd.h>
#include <sys/mman.h>
void *mmap( void *start, size_t length,
            int prot, int flags, int fd, off_t offset );
```

## **munmap函数**

- ✓ munmap函数用来取消参数start所指的映射内存起始地址，参数length则是欲取消的内存大小。
- ✓ 当进程结束或利用exec相关函数来执行其他程序时，映射内存会自动解除，但关闭对应的文件描述符时不会解除映射。成功返回0，否则返回 - 1，错误原因存于errno中。

```
#include<unistd.h>
#include<sys/mman.h>
int munmap(void *start,size_t length);
```

## fstat函数

- ✓ fstat函数用来将文件描述符fildes所指的文件的的状态，复制到参数buf所指的  
结构中（struct stat）。执行成功则返回0，失败返回-1，错误代码存于  
errno。

```
#include<sys/stat.h>
#include<unistd.h>
int fstat(int fildes, struct stat *buf);
```

## struct stat结构

```
struct stat
{
    dev_t  st_dev; /*文件的设备编号      */
    ino_t  st_ino; /*文件的inode      */
    mode_t st_mode; /*文件的类型和存取的权限  */
    nlink_t st_nlink; /*连到该文件的硬连接数目  */
    uid_t  st_uid; /*文件所有者的用户识别码  */
    gid_t  st_gid; /*文件所有者的组织别码    */
    dev_t  st_rdev; /*若此文件为装置设备文件, 则为其设备编号 */
    off_t  st_size; /*文件大小, 以字节计算      */
    unsigned long st_blksize; /*文件系统的I/O缓冲区大小  */
    unsigned long st_blocks; /*占用文件区块的个数      */
    time_t  st_atime; /* time of last access      */
    time_t  st_mtime; /* time of last modification */
    time_t  st_ctime; /* time of last change      */
};
```



# 嵌入式软件设计

大连理工大学 赖晓晨

