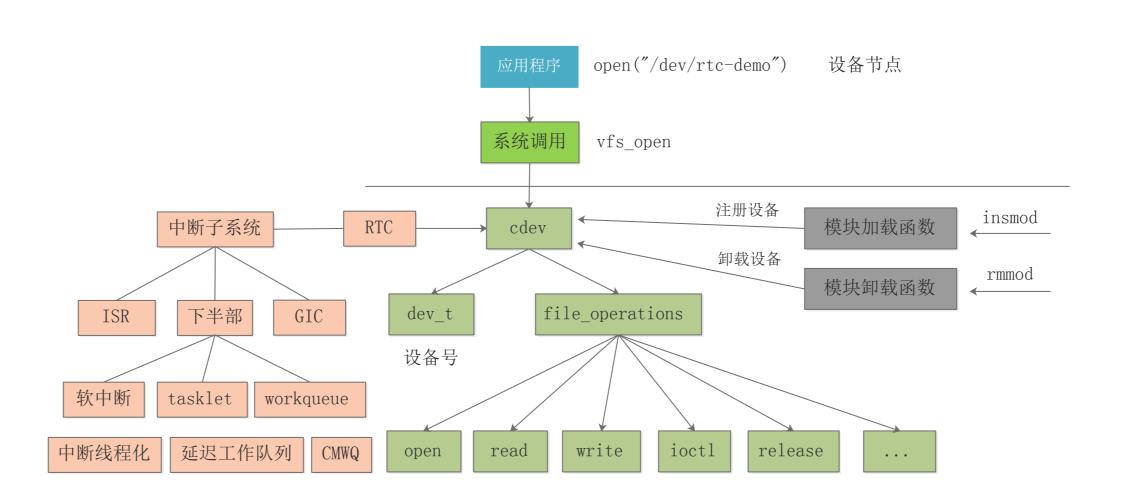
作者博客: www. zhaixue. cc 淘宝店: https://wanglitao. taobao. com

```
open("dev/rtc-demo0", 0 RDWR):
                                                                                                            register_chrdev注册流程
           用户驱动: rtc字符驱动
                                                                                                                                                                           1. 通过系统调用,调用sys_open,根据设备名对应的inode信息i_flags找到设备类型,创建file
                                                                                                                                                                           2. 根据设备号(inode->i_rdev)查找cdev: kboj_lookup
                                                                                                            1. 申请设备号: __register_chrdev_region
          drivers/rtc.c
                                                                                                            2. 创建cdev对象: cdev alloc
                                                                                                                                                                          3. 保存找到的cdev: inode->i_cdev = cdev
           static const struct file_operations rtc_fops = {
                                                                                                                                                                          4. 将inode添加到cdev->list链表中
                                                                                                            3. 初始化cdev对象: cdev->ops = fops
               .owner = THIS MODULE,
                                                                                                                                                                          5.用cdev中的ops初始化file: file->f_op = cdev->ops
                                                                                                            4. 将cdev添加到驱动模型中: cdev_add, kboj、kset
                                                                                                                                                                          6. 调用open回调函数: file->f_op->open(inode, filp)
7. open执行完成,通过VFS返回给用户空间一个文件描述符: fd
                                                                                                            5. 将cdev和chrdevs数组中的char_device_struct建立关联: cd->cdev = cdev
               . open = rtc open,
               . read = rtc_read,
                                                                                            include/linux/fs.h fs/chr dev.c
                                                                                                                                                                            fd和内核中的file相对应,上层通过fd就可以找到file,找到对应的read、write函数
               .write = rtc_write,
                                                                                            static inline int register chrdev (unsigned int major, const char *name,
                                                                                                                                                                           fs/inode.c:
               .unlocked_ioctl = rtc_ioctl,
                                                                                            □□□□ const struct file operations *fops)
                                                                                                                                                                            void init special inode(struct inode *inode, umode t mode, dev t rdev)
               .release = rtc release,
                                                                                            □return __register_chrdev(major, 0, 256, name, fops);
                                                                                                                                                                            □inode->i mode = mode;
                                                                                            int __register_chrdev(unsigned int major, unsigned int baseminor,
                                                                                                                                                                            ☐ if (S ISCHR(mode)) {
           static int __init rtc_init(void)
                                                                                                     unsigned int count, const char *name,
                                                                                                                                                                            \square inode->i fop = &def chr fops;
                                                                                            const struct file_operations *fops)
                                                                                                                                                                            \square inode->i rdev = rdev;
               int ret = 0;
                                                                                                                                                                            □} else if (S ISBLK(mode))
                                                                                            □struct char_device_struct *cd;
                                                                                                                                                                            \square \square inode \rightarrow i_fop = \&def_blk_fops;
              regs = (rtc_reg_t *)ioremap(RTC_BASE, sizeof(rtc_reg_t));
                                                                                            □struct cdev *cdev;
                                                                                                                                                                            \square \square \text{inode} \rightarrow \text{i rdev} = \text{rdev};
              printk("rtc init\n");
                                                                                            \Box int err = -ENOMEM;
                                                                                                                                                                            □} else if (S_ISFIF0(mode))
                                                                                                                                                                            \square inode->i fop = &pipefifo fops;
               ret = register_chrdev(222, "rtc-demo", &rtc_fops); __
                                                                                            Dcd = __register_chrdev_region(major, baseminor, count, name);
                                                                                                                                                                            □else if (S ISSOCK(mode))
               if (ret < 0) {
                                                                                            \Box cdev = cdev_alloc();
                                                                                                                                                                            \square\square;\square/* leave it no open fops */
                   printk("Register char module: rtc failed..\n");
                                                                                            \Box cdev \rightarrow owner = fops \rightarrow owner;
                                                                                                                                                                            □else
                   return 0;
                                                                                            \Box cdev \rightarrow ops = fops;
                                                                                                                                                                            \square \square printk("... \n");
                                                                                            □kobject_set_name(&cdev->kobj, "%s", name);
               else {
                                                                                            □cdev_add(cdev, MKDEV(cd->major, baseminor), count);
                   printk("Register char module: rtc success!\n");
                                                                                            \Box cd \rightarrow cdev = cdev;
                                                                                                                                                                           fs/chr dev.c:
                                                                                                                                                                            const struct file operations def chr fops = {
                                                                                            □return major ? 0 : cd->major;
               return 0;
                                                                                                                                                                            \Box. open = chrdev open,
                                                                                                                                                                            \square. 11seek = noop 11seek,
                                                                                                                                                                           static int chrdev open(struct inode *inode, struct file *filp)
                                                                                                           #define CHRDEV MAJOR HASH SIZE 255
    struct file_operations {
                                                                                                            // 全局数组每个char_device_struct元素对应一个主设备号
        struct module *owner;
                                                                                                                                                                            □const struct file operations *fops;
                                                                                                           static struct char_device_struct
        loff_t (*llseek) (struct file *, loff_t, int);
                                                                                                                                                                            □struct cdev *p, *new;
        ssize_t (*read) (struct file *, char __user *, size_t, loff_t *);
                                                                                                                                                                            \Box p = inode \rightarrow i cdev;
                                                                                                           □struct char_device_struct *next;
        ssize_t (*write) (struct file *, const char __user *, size_t, loff_t *);
                                                                                                                                                                            □□struct kobject *kobj;
                                                                                                            □unsigned int major;
        int (*iterate) (struct file *, struct dir_context *);
                                                                                                                                                                            \square int idx;
                                                                                                            □unsigned int baseminor;
        __poll_t (*poll) (struct file *, struct poll_table_struct *);
                                                                                                                                                                            □□kobj = kobj_lookup(cdev_map, inode->i_rdev, &idx);
                                                                                                            □int minorct;
        long (*unlocked ioctl) (struct file *, unsigned int, unsigned long);
                                                                                                                                                                            \square new = container of (kobj, struct cdev, kobj);
                                                                                                            \Box char name [64];
        long (*compat_ioctl) (struct file *, unsigned int, unsigned long);
                                                                                                                                                                            \square inode->i cdev = p = new;
                                                                                                            □struct cdev *cdev; □/* will die */
        int (*mmap) (struct file *, struct vm_area_struct *);
                                                                                                                                                                            \square list add(&inode->i devices, &p->list);
                                                                                                             *chrdevs[CHRDEV MAJOR HASH SIZE];
        unsigned long mmap_supported_flags;
                                                                                                                                                                             \square \square \text{new} = \text{NULL};
        int (*open) (struct inode *, struct file *);
                                                                                                                                                                             \Boxfops = fops_get(p->ops);
        int (*flush) (struct file *, fl_owner_t id);
                                                                                                                                                                             \exists replace\_fops(filp, fops);
        int (*release) (struct inode *, struct file *);
                                                                                                                                                                             \Boxfilp->f op->open(inode, filp);
                                                                                                                      struct cdev {
        int (*fsync) (struct file *, loff_t, loff_t, int datasync);
                                                                                                                                                                             □return 0;
                                                                                                                          struct kobject kobj;
        int (*fasync) (int, struct file *, int);
                                                                                                                          struct module *owner;
                                                                                                                          const struct file operations *ops;
                                                                                                                         ▼ struct list head list;
                                                                                                                          _dev t dev;
  struct fdtable {
                                                                                                                                                                                       SYSCALL DEFINE3 (open, const char user *, filename, int,
                                                                                                                          unsigned int count;
   □unsigned int max_fds;
                                                                                                                                                                                        flags, umode t, mode)
   □struct file rcu **fd; /* current fd array */
                                                                 struct fd
   □unsigned long *close_on_exec;
                                                                                                                                                                                           if (force o largefile())
                                                                 □struct file *file;
   □unsigned long *open fds;
                                                                                                                                                                                        \Boxflags |= 0 LARGEFILE;
                                                                 □unsigned int flags;
   □unsigned long *full_fds_bits;
                                                                                                                           主设备号12位:次设备号20位
                                                                                                                                                                                         return do sys open(AT FDCWD, filename, flags, mode);
   □struct rcu_head rcu;
                                                                                                                                                                                        long do_sys_open(int dfd, const char user *filename, int
                                                                                                                                                                                        flags, umode t mode)
fs/read write.c
SYSCALL_DEFINE3 (read, unsigned int, fd, char __user *, buf, size_t, count)
                                                                                                                                                                                           struct open_how how = build_open_how(flags, mode);
                                                                                                                mknod: 每创建一个文件,文件系统就会分配一个inode,文件与inode——对应
                                                                                                                                                                                           return do sys openat2(dfd, filename, &how);
□return ksys read(fd, buf, count);
                                                                                                               struct inode { //内核中的所有文件都使用inode来表示
ssize t ksys read(unsigned int fd, char user *buf, size t count)
                                                                                                                   umode_t \square \square i_mode;
                                                                                                                                                                                      do_sys_openat2
                                                                                                                   unsigned short□□i_opflags;
                                                                                                                                                                                           -> do filp open->path openat->do open->vfs open
□struct fd f = fdget pos(fd); //通过描述符fd找到对应的fd结构和file对象
                                                                                                                   kuid_t□□i_uid; // inode拥有者ID
                                                                                                                                                                                                 ->do_dentry_open: file->f_op=inode->i_fop /
□ssize_t ret = -EBADF;
                                                                                                                   kgid_t□□i_gid; // inode所属群组ID
                                                                                                                                                                                                                   file->f op->open
                                                                                                                   unsigned int□□i_flags
                                                                                                                                                     ;//文件类型
                                                                                                                                                                                           -> fd_install(fd, file)->__fd_install:fdt->fd[fd]
\Box if (f. file) {
□□loff_t pos, *ppos = file_ppos(f.file); //获取文件读写位置
                                                                                                                   const struct inode_operations□*i_op;
\square \square \text{ if (ppos)}  {
                                                                                                                   struct super block□*i sb:
\square \square \square pos = *ppos;
                                                                                                                   struct address_space□*i_mapping;
\square \square \square ppos = \&pos;
                                                                                                                   unsigned long□□i_ino;
                                                                                                                                                                                       struct file { //用来表示进程中打开的文件
                                                                                                                   union {
\Boxret = vfs_read(f.file, buf, count, ppos);
                                                                                                                                                                                       □union {
                                                                                               多个设备可以共用同 □ □ □ □ □ □ □ □ □ □ □ □ ;
                                                                                                                    const unsigned int i_nlink;
                                                                                                                                                                                       □ struct llist node □ fu llist;
\Box \Box if (ret \geq 0 \&\& ppos)
                                                                                                                     unsigned int __i_nlink;
                                                                                                                                                                                       □ struct rcu_head □fu_rcuhead;
\Box\Box\Box f. file \rightarrow f_pos = pos;
                                                                                                                                                                                       \square} \f_u;
\Box \Box fdput_pos(f);
                                                                                                                   dev_t \square \square i_r dev;
                                                                                                                                                     // 若是设备文件,表示设备号
                                                                                                i_devices 和cdev中
                                                                                                                                                                                       □struct path□□f_path;
                                                                                                的list组成一个链表
                                                                                                                   loff t \square \square i size;
                                                                                                                                                                                       □struct inode□□*f inode;□
□return ret;
                                                                                                                   unsigned short
                                                                                                                                            \Box i_bytes;
                                                                                                                                                                                       □const struct file_operations□*f_op;
                                                                                                                   u8\square\square i\_blkbits;
                                                                                                                                                                                       \squarespinlock t\square \squaref lock;
                                                                                                                   u8□□□i_write_hint;
ssize_t vfs_read(struct file *file, char __user *buf, size_t count, loff_t *pos)
                                                                                                                                                                                       \square enum rw_hint \square \square f_write_hint;
                                                                                                                   blkcnt_t□□i_blocks;
                                                                                                                                                                                       \square atomic long t\square \square f count;
                                                                                                                                                                                       \squareunsigned int \square \squaref_flags;
□ssize_t ret;
                                                                                                                   unsigned long□□dirtied_when;□
                                                                                                                                                                                       \Boxfmode t\Box \Box f mode;
                                                                                                                   unsigned long □ dirtied_time_when;
□ret = rw_verify_area(READ, file, pos, count);
                                                                                                                                                                                       void□□□*private_data
                                                                                                                   union {
☐ if (count > MAX RW COUNT)
                                                                                                                                                                                       \squareerrseq t\square\squaref wb err;
                                                                                                               □ struct hlist_head□i_dentry;
\square count = MAX_RW_COUNT;
                                                                                                                                                                                       □errseq_t□□f_sb_err; /* for syncfs */
                                                                                                                    struct rcu head□i rcu;
                                                                                                               \square};
\Box if (file->f_op->read)
                                                                                                                   atomic64_t\squarei_version;
ret = file->f op->read(file, buf, count, pos);
                                                                                                                   atomic64_t\squarei_sequence;
□else if (file->f_op->read_iter)
                                                                                                                   atomic_t\square \square i_count;
                                                                                                                                                                                       include/linux/fdtable.h
\square ret = new sync read(file, buf, count, pos);
                                                                                                                                                                                       struct files_struct { // 用户进程打开的文件列表
                                                                                                                   atomic_t□□i_dio_count;
□else
                                                                                                                   atomic_t□□i writecount;
                                                                                                                                                                                              atomic t count;
\Boxret = -EINVAL;
                                                                                                                                                                                              bool resize_in_progress;
□inc_syscr(current);
                                                                                                                                                                                              wait_queue_head_t resize_wait;
                                                                                                                   union {
□return ret;
                                                                                                               □ const struct file_operations *i_fop;□
                                                                                                                    void (*free inode) (struct inode *);
                                                                                                                                                                                              struct fdtable rcu *fdt;
                                                                                                               □};
                                                                                                                                                                                              struct fdtable fdtab;
                                                                                                                                                                                              spinlock_t file_lock ____cacheline aligned in smp;
                                                                                                                  struct address_space□i_data;
                                                                                                                                                                                              unsigned int next fd;
                                                                                                                  \ struct list_head□i_devices;
                                                                              i_cdev为 NULL,说明该设备文件没有被打开
                                                                                                                                                                                              unsigned long close on exec init[1];
                                                                                                                  union {
                                                                                                                                                                                              unsigned long open fds init[1];
                                                                                                                     struct pipe_inode_info *i_pipe;
                                                                                                                     struct block device *i bdev;
                                                                                                                                                                                              unsigned long full fds bits init[1];
                                                                                                                struct cdev□ *i_cdev; //指向对应的cdev
                                                                                                                                                                                        struct file __rcu * fd_array[NR_OPEN_DEFAULT];
                                                                                                                                  *i_link;
                                                                                                               \Box char \Box
                                                                                                               \square unsigned \square i_dir_seq;
                                                                                                               \square};
                                                                                                                     \underline{\quad} u32\square \square i_generation;
                                                                                                                   void□□□*i private;
                                                                                                                                                                                                File Descriptor: fd
                                                                                                                                                                                                 fd = open("/dev/rtc-demo0", 0_RDWR);
```



淘宝店: https://wanglitao.taobao.com