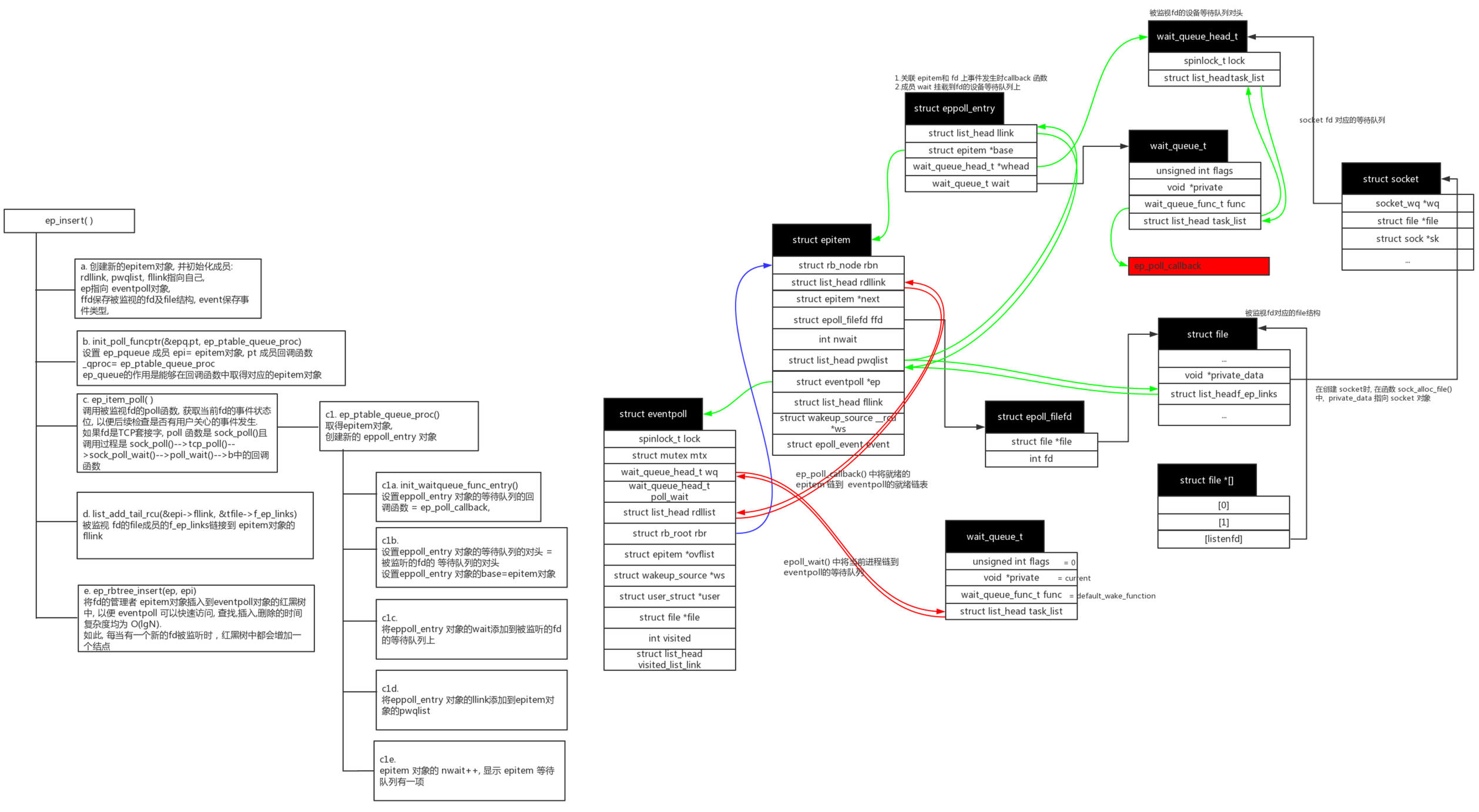
ep_insert( )
a. 创建新的epitem对象, 并初始化成员:
rdllink, pwqlist, fllink指向自己,
ep指向 eventpoll对象,
ffd保存被监视的fd及file结构, event保存事件类型,
b. init_poll_funcptr(&epq.pt, ep_ptable_queue_proc)
设置 ep_pqueue 成员 epi= epitem对象, pt 成员回调函数 _qproc= ep_ptable_queue_proc
ep_queue的作用是能够在回调函数中取得对应的epitem对象
c. ep_item_poll( )
调用被监视fd的poll函数, 获取当前fd的事件状态位, 以便后续检查是否有用户关心的事件发生.
如果fd是TCP套接字, poll 函数是 sock_poll()且调用过程是 sock_poll()-->tcp_poll()-->sock_poll_wait()-->poll_wait()-->b中的回调函数
c1. ep_ptable_queue_proc()
取得epitem对象,
创建新的 eppoll_entry 对象
c1a. init_waitqueue_func_entry()
设置eppoll_entry 对象的等待队列的回调函数 = ep_poll_callback,
c1b.
设置eppoll_entry 对象的等待队列的对头 = 被监听的fd的 等待队列的对头
设置eppoll_entry 对象的base=epitem对象
c1c.
将eppoll_entry 对象的wait添加到被监听的fd的等待队列上
c1d.
将eppoll_entry 对象的llink添加到epitem对象的pwqlist
c1e.
epitem 对象的 nwait++, 显示 epitem 等待队列有一项
d. list_add_tail_rcu(&epi->fllink, &tfile->f_ep_links)
被监视 fd的file成员的f_ep_links链接到 epitem对象的fllink
e. ep_rbtree_insert(ep, epi)
将fd的管理者 epitem对象插入到eventpoll对象的红黑树中, 以便 eventpoll 可以快速访问, 查找,插入,删除的时间复杂度均为 O(lgN).
如此, 每当有一个新的fd被监听时，红黑树中都会增加一个结点
被监视fd的设备等待队列对头
wait_queue_head_t
spinlock_t lock
struct list_headtask_list
1.关联 epitem和 fd 上事件发生时callback 函数
2.成员 wait 挂载到fd的设备等待队列上
struct eppoll_entry
struct list_head llink
struct epitem *base
wait_queue_head_t *whead
wait_queue_t wait
wait_queue_t
unsigned int flags
void *private
wait_queue_func_t func
struct list_head task_list
socket fd 对应的等待队列
struct socket
socket_wq *wq
struct file *file
struct sock *sk
...
struct epitem
struct rb_node rbn
struct list_head rdllink
struct epitem *next
struct epoll_filefd ffd
int nwait
struct list_head pwqlist
struct eventpoll *ep
struct list_head fllink
struct wakeup_source __rcu *ws
struct epoll_event event
ep_poll_callback
被监视fd对应的file结构
struct file
...
void *private_data
struct list_headf_ep_links
...
在创建 socket时, 在函数 sock_alloc_file()中, private_data 指向 socket 对象
struct eventpoll
spinlock_t lock
struct mutex mtx
wait_queue_head_t wq
wait_queue_head_t poll_wait
struct list_head rdllist
struct rb_root rbr
struct epitem *ovflist
struct wakeup_source *ws
struct user_struct *user
struct file *file
int visited
struct list_head visited_list_link
ep_poll_callback() 中将就绪的epitem链到 eventpoll的就绪链表
struct epoll_filefd
struct file *file
int fd
struct file *[]
[0]
[1]
[listenfd]
epoll_wait() 中将当前进程链到eventpoll的等待队列
wait_queue_t
unsigned int flags = 0
void *private = current
wait_queue_func_t func = default_wake_function
struct list_head task_list