

1: copyin() securely copies LEN bytes from user-level space to a kernel-level space, copyout() securely copies LEN bytes from kernel-level space to a user-level space

2: UIO_SYSSPACE is used when data transfer is completely within kernel space(i.e kernel I/O)

The difference is UIO_USERISPACE indicates that user-level memory segment being referenced is executable while UIO_USERSPACE is not.

3: we have no access to the file node,so file cannot be closed after going to user mode(md_usermode() does not return)

4: md_usermode()

5: it is a pointer points to a one byte structure of user address space, so it won't mix with other pointers

6: it currently panics. We dont wanna crash the kernel when killing a thread. we must free the resource of the killed thread and let other thread run

7: interrupts are enabled when mips_syscall() is invoked. When kill_curthread() is invoked, the interrupts could be on or off. It depends on the previous context

8: vfs_open()

9: vop_open

vop_close

vop_reclaim

vop_read

vop_readlink

vop_getdirent

vop_write

vop_ioctl

vop_stat

vop_gettype

vop_tryseek

vop_fsync

vop_mmap

vop_truncate

vop_namefile

vop_creat

vop_symlink

vop_mkdir

vop_link

vop_remove
vop_rmdir
vop_rename
vop_lookup
vop_lookupparent

We dont need to create two vnodes.