

## Assignment 2 Question 1 Problems

### **Group 10- cmkatigb, jkpmok, dhdkim**

1) Copyout() copies a block of memory of size LEN bytes from the kernel space address SRC to the user space address USERDEST.

2) The reason why we want to vfs\_close() is because we want to ensure that while in the user space, processes do not have the ability to modify any files the kernel was working on. If a file was modified, then when we revert back to kernel mode, the kernel will be dealing with a different file which we are trying to avoid.

3) enter\_new\_process()

4) Userptr\_t is a pointer to a one-byte struct, so it won't mix with other pointers. It will keep track of the user level address.

5) We must change kill\_curthread() because all it does right now is print out an error statement. We need to implement handling the exception so we can kill the process. It will need to free the thread's memory.

6) OFF for kill\_curthread, OFF for syscall because the operating system is going into kernel mode.

7) Copyin() copies a block of memory with length LEN from user-level address USERSRC to the kernel address DEST. Copyinstr copies a string from user-level address USERSRC to kernel address DEST. Copyinstr utilizes a function called copystr which just copies a null-terminated string from SRC to DEST.

8) Vfs\_open() is used to open a file or device and obtain a vnode.

9) Some operations we can do include: vnode\_cleanup (destroys node), vnode\_check(determines if structure is valid), vnode\_incref, vnode\_decref (increment/decrement counter respectively), vnode\_incopen, vnode\_decopen (increment/decrement open counter respectively).

2 processes can open the same file, however one vnode is created. The reference counter is incremented if this is the case.