Assignment 2 Code Review

Question 1:
The function "copyin", with prototype "int copyin(const_userptr_t usersrc, void *dest, size_t len)", copies a block of memory of length LEN bytes from a user-level address USERSRC to a kernel-space address DEST. It is used to check if a memory address provided by the user-level code is valid to prevent fatal kernel memory faults, since it will return EFAULT if a memory addressing error was encountered.

Question 3:
At the beginning of runprogram, vfs_open is called, which allocates resources for the corresponding file. vfs_close frees any resources the file is using, which must be called before going to user mode, or else the allocated resources will never get freed.

Question 4:
The kernel function that go to user mode after loading an executable is enter_new_process(), which is called by runprogram.c. It calls mips_usermode() which is the function for entering user mode.

Question 5:
Userptr_t is a pointer that points to an address that exists in the userspace.

Question 6:
It should be changed because rather than panic (what it is doing right now) it should clean up zombie children, release/destroy lock, close file descriptors associated with curthread to deallocate resources curthread is using before calling thread_exit.

Question 7:
Interrupts are enabled during both syscall() and kill_curthread().

Question 8:
Copyinstr has prototype "int copyinstr(const_userptr_t usersrc, char *dest, size_t len, size_t *got);". It copies a null-terminated string of at most LEN bytes from a user-space address USERSRC to a kernel-space address DEST. It contains one more parameter (a pointer) than Copyin which is used to store the actual length of string.

Question 9:
The kernel function vfs_open() is used to open a file or device and obtain a

vnode.

Question 10:
Vnode's operations include: open, close, reclaim, read, readlink, getdirentry, write, ioctl, stat, gettype, tryseek, fsync, mmap, truncate, namefile, creat, symlink, mkdir, link, remove, rmdir, rename, lookup, and lookparent. There is also reference count manipulation VOP_INCREF and VOP_DECREF, as well as open count manipulation VOP_INCOPEN and VOP_DECOPEN. (More can be found in vnode.h)
When two different processes open the same file, the vnode's vn_refcount field will be increased, instead of creating two vnodes.