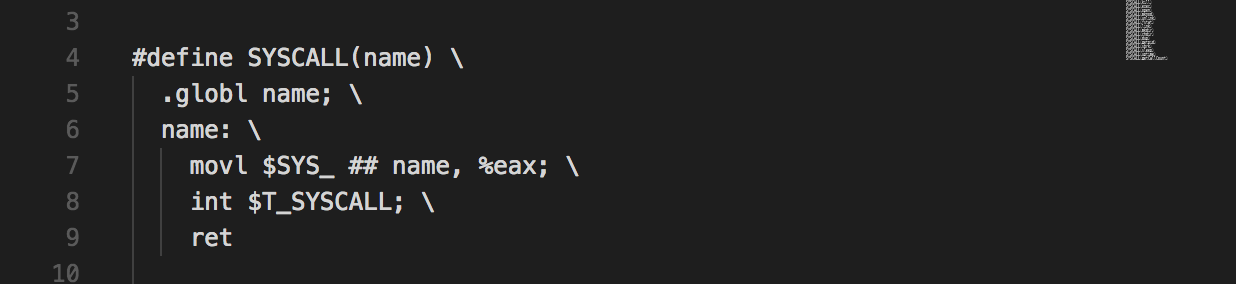
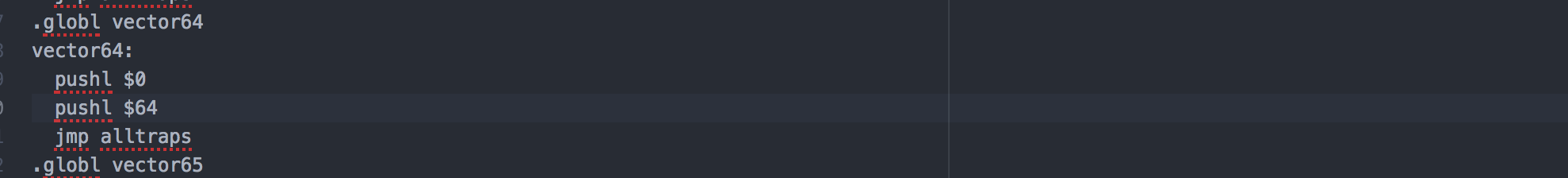
**DESCRIBING THE EXECUTION OF WRITE() SYSTEM CALL**

When the write system call (with value 16) is called, the user process would then execute the system call by first going to **usys.S**.

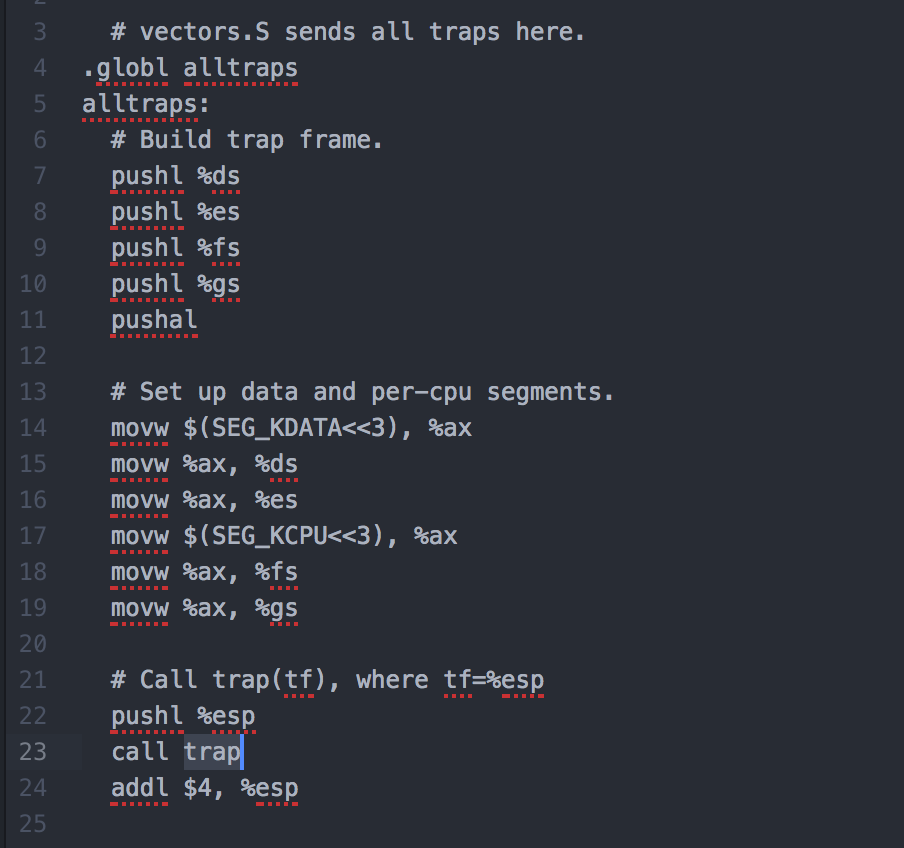


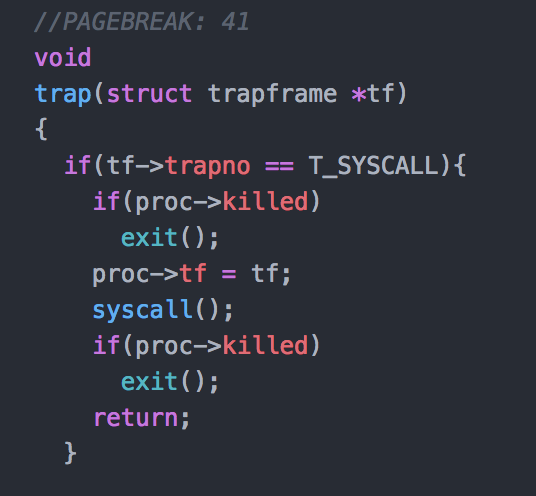
../../../../Desktop/Screen%20Shot%202018-02-28%20at%2011.10.06%20../../../../Desktop/Screen%20Shot%202018-02-28%20at%2011.11.22%20

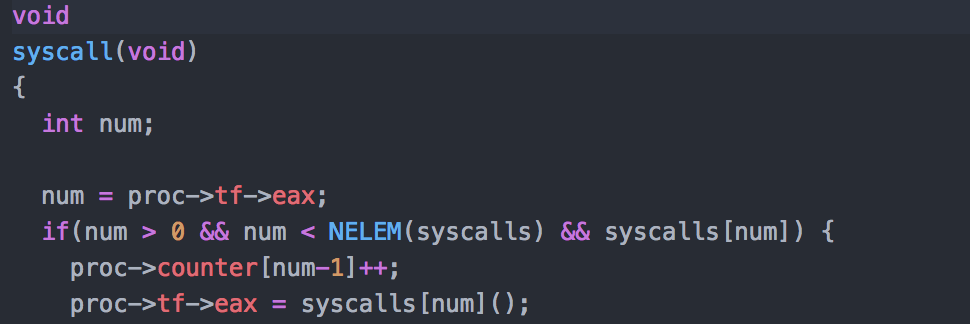
**usys.S** would then get the write system call number defined in **syscall.h** and save it to **eax**. Then it would set the value of interrupt instruction to T\_SYSCALL, which is 64 (getting from **traps.h**). The interrupt would then go to **vectors.S** to execute the TRAP instruction.



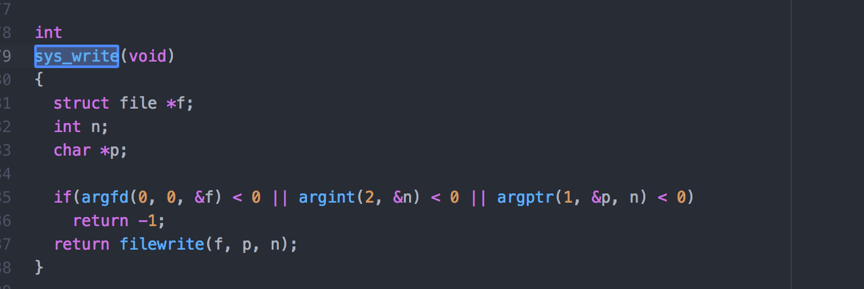
In this handler, in the last line, the code would then jump to run **alltraps** (line “jmp alltraps”) to execute the trap instruction. **alltraps** is in **trapasm.S.**

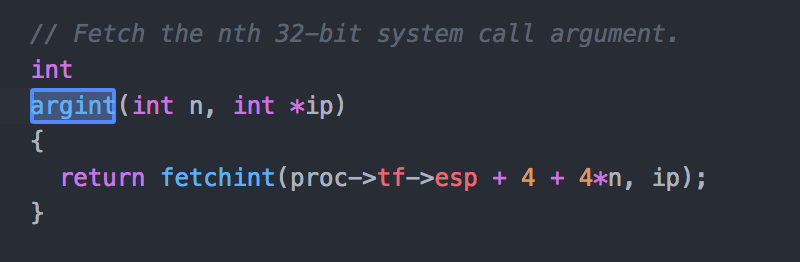
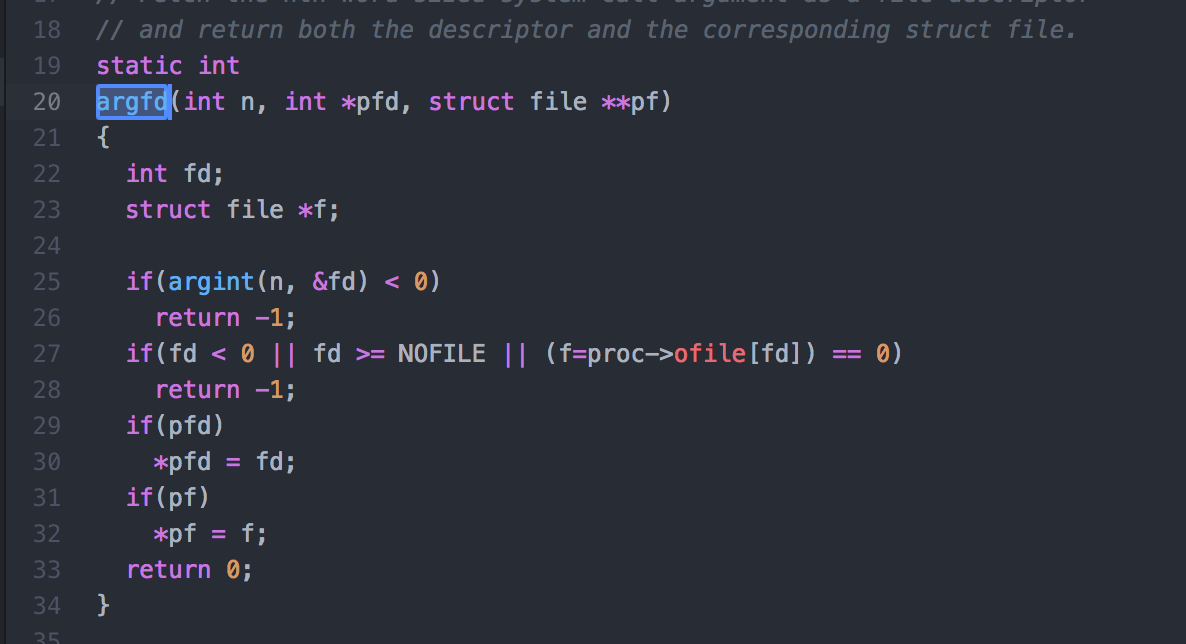


After building the trap frame and setup data, the code will then run the **trap** function in file **trap.c.** 

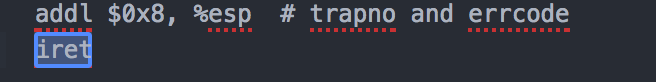


In this function, because it is a system call, it will go into the **if** block. In this function, the trap frame of the process will be the trap frame constructed in **trapasm.S**. Then **syscall()** function in **syscall.c** will be executed. In here, the value 16 in **eax** will be used as the system call number.





Then the system call will be executed and in this case, **sys\_write()** in **sysfile.c.** Tocheck the validity of fd, the function calls **argfd()** in the **if** block, the function would be then call **argint()** to check if the **fd** is provided. In this case it is not so it returns -1. The **sys\_write()** then return -1 to **syscall()**. The variable **proc->tf->eax** will then save the value **-1**.



After that, it will return to the **trap()** function. The **trap()** function would then continue and return to **trapasm.S.** Becausebefore jmp or call, the assembly code has already push the necessary data to the stack, **trapasm.S** would then continue to execute later lines and then return to **usys.S** when reaching **iret** by popping the stack**.** In usys.S, having finished executing the interrupt **int,** it would then return to user space and raise an exception.