**Group 34**

Robert Turza, Hoe Suh Jeong, WonYong Lee, Woo Young Choi

**Netid**: rturza2, wlee94, hjeong13, wchoi19

**Checkpoint 1**

**Bug 1 :** Booting did not work properly

**Date**: Oct, 20, 2018 by everyone

**Detail**: The program was kept rebooting.

**How**: Went through provided reading materials and learned proper way to write rebooting and how to properly set up GDT.

**Solution**: We added “lgdt gdt\_desc\_ptr” code into boot.S which we were missing earlier.

**Bug 2** : Paging did not work

**Date**: Oct, 20, 2018 by Robert

**Detail**: Page Fault Exception occurred

**How**: Went through provided reading materials and read online resources to learn how to write gcc inline assembly

**Solution**: We didn’t properly initialize page directory and page table in page\_init function. We fixed syntaxes for gcc inline assembly codes and properly set bits for page tables. (e.g. “USER\_SUPERVISOR\_SET”…)

**Bug 3** : RTC did not work properly

**Date**: Oct, 21, 2018 by Hoesuh

**Detail**: RTC funtion didn’t work properly that didn’t pass the test in test.c

**How**: Asked TA for the problem

**Solution**: We forgot to put cli(), sti().

**Checkpoint 2**

**Bug 4**: File read did not working correctly

**Date**: Oct, 26, 2018 by Woo

**Detail**: Can’t “file read” properly if it is called multiple times

**How:** Carefully read the code again and discussed with teammates how to fix

**Solution**: read data function takes 4 arguments ( inode index, file position, buffer , length in bytes) and we always called the function with 0 for file position because it always started from the beginning. However, when we run with files with multiple times with splitted length it doesn’t work because we didn’t update the file position. We changed it to “offset” variable and updated the file position.

**Bug 5**: Terminal read did not working correctly

**Date**: Oct, 27, 2018 by Robert

**Detail**: Terminal read should wait until enter key is pressed but our code didn’t.

**How:** We got points off from the check point 2 demo and the TA told us what is the problem with Terminal Read function.

**Solution**: We implemented get\_enter\_key function in keyboard.c and implemented while loop in terminal read function which waits until enter is pressed.

**Checkpoint 3**

**Bug 6:** Not reading the correct argument data

**Date:** Nov, 11, 2018 by Woo

**Detail**: We could not pass correct argument data from execute function.

**How**: We used prinf statement to print out all the data stored in argument data and counted the number of each characters

**Solution**: We used if statement to check more boundary conditions, such as “ “(space), “\0”, and “\n”(newline).

**Bug 7**: Shell was not operating properly

**Date**: Nov, 11, 2018 by Hoesuh

**Detail**: After launching shell process, the program gives page fault, and could not see any output on the terminal.

**How:** We printed out bunch of printf statement on terminal before it reaches the page fault. By using this numerous printf statement, we were able to figure out the exact function or functionality was causing errors.

**Solution**:

**Bug 8**: could not find dentry error

**Date**: Nov, 11, 2018 by Robert

**Detail**: If we use print test and other functions provided for us, terminal states that it could not find dentry. Meaning it cannot find file same filename within filesystem.

**How:** We printed out the argument we are comparing and compared with the actual filename that the program seek to find.

**Solution**: We parsed through the argument with different amount of characters. With one filename has one extra character at the end made the boundary conditions to fail resulting in “dentry not found.” We fixed the amount of character parsing.

**Bug 9**: Halt did not work properly

**Date**: Nov, 12, 2018 by Wonyong

**Detail**: Halt function was not working when we exit from the shell.

**How:** We asked TA for help to locate the problem

**Solution**: We added parent pcb to keep track of original pcb. After this, function was able to go back to previous state which is shell.

**Checkpoint 4**

**Bug 10**: Grep wasn’t operating properly

**Date**: Nov, 25, 2018 by Robert

**Detail**: Grep keeps printing something even with no argument.

**How** : We revised the read\_data code in filesystem.c

**Solution**: We figured out our file position were not reseted after reading a certain file, and thus,

revised to have it reset every time opening new file

**Bug 11**: Hello wasn’t operating properly

**Date**: Nov, 25, 2018 by Robert

**Detail**: Grep keeps printing something even with no argument.

**How**: We revised the read\_data code in filesystem.c

**Solution**: We figured out our file position were not reseted after reading a certain file, and thus,

revised to have it reset every time opening new file

**Bug 12**: Fish wasn’t working correctly

**Date**: Nov, 26, 2018 by Woo

**Detail**: The Fish just wasn’t working correctly.

**How**: We got points off during Demo because fish didn’t work properly and we realized that our vidmap function and paging were not correct.

**Solution**: There also was a problem with file\_read. We stored inode index at pcb structure and used that inode index for file\_read function. Also, fixed the arguments. It now sets the 0th character to ‘\0’ in the argument when the program halts.

**Bug 13:** Cat file didn’t work properly

**Date**: Nov, 26 , 2018 by Wonyong

**Detail**: cat verylargetextwithverylongname.txt wasn’t working properly. We should be enable to read when we type “cat verylargetextwithverylongname.txt” and able to read when we type “cat verylargetextwithverylongname.tx”

**How**: We did boundary check by checking every single input parsed characters.

**Solution**: We changed the size of the array and changed the method of parsing that enables better boundary check.

**Checkpoint 5**

**Bug 14**: Unable to load three shells at the beginning of the program

**Date**: Dec, 7, 2018 by Hoesuh

**Detail**: Could not load three shells at the beginning of the kernel because shell never returns.

made 3 PCBs during the initializations, and then actually called only one shell.

**How**: Switching between terminals shows same screen.

**Solution**: When the PIT ticks, execute shell, until we open up all three terminals.

**Bug 15**: Keyboard not working on switched terminal(1)

**Date**: Dec, 8, 2018 by Woo

**Detail**: Stuck on switched terminal

**How**: We couldn’t type on the terminal and using gdb we were stuck on terminal\_read..

**Solution**: We were sending eoi at the end of the keyboard handler. By setting, send eoi at the beginning of the keyboard handler solved this problem.

**Bug 16**: Keyboard not working on switched terminal(2)

**Date**: Dec, 8, 2018 by Wonyong

**Detail**: Stuck on switched terminal

**How**: We couldn’t type on the terminal and using gdb we were stuck on terminal\_read..

**Solution**: We have set up the pit, however as before we forgot to send eoi.

**Bug 17**: Random background color for terminal 2

**Date**: Dec, 9, 2018 by Robert

**Detail**: Terminal 2 shows green and black dots everywhere.

**How**: Tried to debug, but didn’t know why. However, we were confident about the code except for the paging.

**Solution**: We have used the wrong paging function, re-wrote paging.c trying to understand paging.

**Bug 18**: Fish on terminals other than the terminal that we ran

**Date**: Dec, 9, 2018 by Wonyong

**Detail**: Fish should not show up on other terminals, but it did.

**How**: Asked the guy next to us working on 391 in engineering hall.

**Solution**: We had to page correctly to the screen, which used vidmap. After doing scheduling we were good without the solution.

**Bug 19**: 3 Terminals are page faulting frequently and video memory isn’t swapping correctly.

**Date**: Dec, 9, 2017 by Hoesuh

**Detail**: After we thought we were done with the scheduling everything started to break apart.

**How**: Running ls on three terminals started to page fault, and the cursor wasn’t working as we thought it would.

**Solution**: Was not even sure that we were running three terminals, so rather than booting up three terminals at the initalization, we have decided to open up the terminals as we press ALT + F2, ALT + F3. So, we restarted the CP5. Also we have decided to do 1:1 vid map paging so that it can be easier to code.

**Bug 20**: Triple Faulting

**Date**: Dec, 10, 2018 by Robert

**Detail**: As we were running the new code with scheduling that we had before, it started to triple fault.

**How**: The program rebooted some times and even break the test\_debug.

**Solution**: Hoesuh’s initialization for the new code was wrong. He have set the current process as -1. However, uint8\_t makes -1 as 255, creating triple faulting while we are calculating pcb addresses.