Checkpoint	Bug	Description	Fix
Checkpoint 1	No interrupts worked	Even when we enabled interrupts, no interrupts were being received from the keyboard or RTC because PIC was initialized incorrectly.	Rewrote PIC initialization code to correctly initialize PIC.
Checkpoint 1	Paging any memory page faults	When trying to page memory within the bounds of the virtual memory, the operating system still page faulted.	Rewrote paging code.
Checkpoint 1	Keyboard stops printing to screen after first letter	The keyboard would print the first key pressed but then the entire kernel would freeze after.	Rewrote the wrappers for the keyboard and rtc. Registers weren't being saved properly and the keyboard handler couldn't return.
Checkpoint 1	Cursor not moving	The cursor does not position itself correctly on the screen with typing	Wrote a function to update the cursor with the VGA port so that the cursor updates correctly
Checkpoint 2	Pressing multiple keys bug	Pressing multiple keys and then releasing one would occasionally release the wrong key. This includest SHIFT, CAPS, and CTRL.	Rewrote code that checked the release key mask and key released. Added more flags for the shift, caps, and ctrl keys.
Checkpoint 2	Reading file system returns all null characters	When calling read data on any files, the buffer would be filled with null characters.	We were somehow only reading from one position in the file multiple times. Rewrote the read data function to read all data blocks correctly.
Checkpoint 2	Buffer overflow would cause page fault	Writing past the buffer limit would cause a page fault.	Incorrectly set buffer limits. Created constants for buffer limits.
Checkpoint 2	RTC write didn't change frequency	Executing rtc write with different frequencies wouldn't change the rate of the RTC.	Rewrote logic to convert frequency to interrupt rate.
Checkpoint 2	Text wrap around in terminal caused strange results	Text wrap around would cause text to appear in random places on the screen.	Rewrote code that tracked the x and y values for the starting position of the screen. Had multiple random variables and removed some of those for consistency.
Checkpoint 2	Reading a directory would return same file	Opening a directory and continuously reading would return the same file name over and over.	Rewrote global variable to keep track of file position. Eventually moved this variable to the corresponding file descriptor file_position variable.
Checkpoint 3	Execute would instantly cause general protection error	Executing any program would cause a general protection error.	Rewrote the system call wrapper as the wrapper was destroying the stack. Fixed our TSS set up and stack set up for the iret in execute. In addition, we were copying the PCB into the wrong spot in memory.
Checkpoint 3	Halt would cause a page fault	Attempting to halt any program would instantly cause a page fault.	Rewrote code that restored past EBP, ESP, and realized we were setting up the TSS incorrectly.
Checkpoint 3	Halt would cause all programs to halt	Calling halt on a child program would call all programs up to the parents to halt as well.	Moved our ESP and EBP saves into an array to prevent halt from continuing down the stack.
Checkpoint 3	Program loader was loading garbage	When stepping through GDB, could see that the loader wasn't loading the program correctly.	Updated read data and the way it handled offsets.
Checkpoint 3	Ping pong interrupts were not working	Ping pong required the user to hold down enter to work.	Enabled interrupts at the beginning of every RTC read.
Checkpoint 4	grep command not working correctly	After doing cat on files, immediately doing a grep command afterwards would not return anything	The function that retrieved arguments from the command incorrectly parsed the commands and remnants of the last command would appear in the arguments for the second command and interfere with the grep. Added a few lines of codes that correctly identified the end of string of arguments.
Checkpoint 4	fish command not working completely	Calling fish would only print out two characters on the terminal	The read_data function in filesystem wasn't implemented correctly and did not correctly account for offset, which fish uses to display the fish. We rewrote the read_data function.
Checkpoint 4	syserr call would not pass all tests	Test 6 would fail when attempting to test with an erroneous vidmap error	Bounds on our vidmap error vlues were incorrect so we corrected the bounds to the correct values
Checkpoint 5	New terminals not accepting keyboard interrupts	After switching terminals, no keyboard interrupts were being received by the new terminal and as such, we could not type anything	Disabled keyboard interrupts before doing a terminal switch and then re-enabled them in the new instance of shell running in the new terminal
Checkpoint 5	Context switching between terminals not working correctly	We can switch between terminals correctly, but when we type in one terminal, switch to another terminal then switch back, we get a page fault	We had multiple page directories set up for each program, but were switching to the wrong page directory when changing terminals. Updated the code to change to the correct page directory.
Checkpoint 5	Line buffer is not in the correct location	When we type a command in one terminal, switch to another terminal, then switch back without typing anything in the second terminal, the characters appear one line above where it is supposed to be	Tweaked the code that set the start screen position in the keyboard handler to update on terminal switch.