ECE 391 BUG LOG

TEAM XI

Summary	Description	Date created	Date modified	Milestone	Status
Ctrl + L breaks cursor position with partially-filled buffers/tab history.	To reproduce: type. Ctrl+L.	4/29/2012 15:06	4/30/2012 16:24	Checkpoint 4	closed
Tab completion occasionally loses cursor position.	Tab completion occasionally gets out of sync with everything else. To reproduce: enter any command with arguments (cat frame0.txt, grep /, whatever). Backspace to remove the space from the expression. Tab complete.	4/29/2012 14:53	4/30/2012 16:48	Checkpoint 4	closed
printing a newline in a background terminal doesn't work	it looks like it just keeps re-writing the same line it works partially since the shell prompts work when printed in background to reproduce: run sched with 10000 and then switch terminals; when you come back sched will be complete but will look just like it did when you left	4/28/2012 21:29	4/30/2012 16:45	Checkpoint 4	closed
scrolling in background terminal doesn't do anything	may not actually be called anymore due to issue #20	4/28/2012 20:25	4/30/2012 16:45	Checkpoint 4	closed
Loading filesystem from disk	I've checked over the image both with xxd and in gdb in the kernel and everything seems to be in the correct place. But it mysteriously doesn't work correctly. Something strange happens inside of shell and it comes back.	4/28/2012 10:40	4/30/2012 16:43	Checkpoint 4	closed
sched is run almost entirely with interrupts disabled	makes it impossible to switch terminals while it's running and demonstrate background printing	4/27/2012 22:10	4/30/2012 16:41	Checkpoint 4	closed

ECE 391 BUG LOG

TEAM XI

gray cursors are left all over the place		4/27/2012 22:08	4/30/2012 16:39	Checkpoint 4	closed
scrolling in a background terminal affects everything	need to scroll only in terminal that caused scrolling	4/27/2012 22:08	4/30/2012 16:38	Checkpoint 4	closed
cursor does not update when printing in background	even though the prompt is printed, when you return to the shell that position is lost or never stored	4/27/2012 22:05	4/30/2012 16:37	Checkpoint 4	closed
entering an invalid command in shell freezes the shell process	similar to what was happening before when getting back to a process that had previously been context switched (eg, something like hello that requires user input)	4/27/2012 1:59	4/30/2012 16:32	Checkpoint 4	closed
sound does not correctly transition over blocks	some sound is lost in moving from one block to another	4/25/2012 21:27	4/30/2012 16:29	Checkpoint 4	closed
sb16 does not generate interrupts		4/25/2012 21:25	4/30/2012 16:33	Checkpoint 4	closed

ECE 391 BUG LOG

TEAM XI

Save/Restore and Returning from SysCall	I saved and restored registers, but I'm not really sure what else needs to be saved. I found varying resources as to how the stack should be set up before iret. I think what I have works, but I'm not really sure how to test it. I also found a discrepancy: According to what I've read, for Linux the segment selector values are as follows: KERNEL_CS 0x0008 KERNEL_DS 0x0010 USER_CS 0x0018 USER_DS 0x0020 While in x86_desc.h they are: KERNEL_CS 0x0010 KERNEL_DS 0x0018 USER_CS 0x0023 USER_DS 0x002B	4/13/2012 4:29	4/15/2012 12:04	Checkpoint 3	closed
Inclusion Guards	Noticed some header files didn't have inclusion guards. Things work now, but someone should add them to prevent future bugs that may result from multiple inclusion.	3/30/2012 2:54	3/30/2012 7:42	Checkpoint 2	closed
Cursor position incorrect while inserting in middle of line	This occurs because the screen coordinates are left at the end of the line after re-printing the buffer. We need to update them to the current position within the buffer if we're not at the end (`keyboard_buffer_pos < keyboard_buffer_size`).	3/27/2012 7:52	4/30/2012 16:29	Checkpoint 2	closed

ECE 391 BUG LOG

TEAM XI

5.2.4 Checkpoint 2 handin (preparations)	You will need to demonstrate that your open, read, and write functions for the three device drivers work correctly. This functionality is independent of how your operating system may use the devices, but it is a good idea for you to start thinking about how you want to interface these functions with the corresponding system calls. You will need to show that when a key is pressed, the keyboard driver stores the corresponding letter in a buffer and that by explicitly calling the keyboard read function you can receive the correct letters and print them out on the screen. Similarly, you will need to demonstrate that you can change the rate of the RTC clock using reads and writes. A good way of doing so is to have the RTC interrupt handler increment a counter and write to a specific spot on the screen. Finally, you will need to demonstrate in Linux that you can read from the read-only file system.	3/15/2012 18:30	4/2/2012 23:50	Checkpoint 2	closed
5.2.3 The real-time clock driver	You will need to write the open, read, and write functions for the real-time clock and demonstrate that you can change the clock frequency. You will need to do some research on how the real time clock works and what the device driver needs to do to communicate with it.	3/15/2012 18:30	4/2/2012 23:50	Checkpoint 2	closed

ECE 391 BUG LOG

TEAM XI

5.2.2 Parse the read-only file system	You will need to support operations on the file system image provided to you, including opening and reading from files, opening and reading the directory (there's only one—the structure is flat), and copying program images into contiguous physical memory from the randomly ordered 4 kB "disk" blocks that constitute their images in the file system.	3/15/2012 18:29	4/2/2012 23:50	Checkpoint 2	closed
5.2.1 Create a terminal driver	When printable characters are typed at the keyboard, they should be displayed to the screen. You will need to keep track of the screen location for this purpose. You need not support scrolling (wraparound is ok), but you may want to interpret CTRL-L or some similar non-printable key as meaning "clear the screen and put the cursor at the top" to make your testing experience more pleasant. You do need to support backspace and line-buffered input. That is, when a read system call is made to the terminal, you must only return data after the user presses enter. Until that time, data should be buffered in the driver and edited appropriately (deleted from both screen and buffer when backspace is seen, etc.).	3/15/2012 18:28	4/2/2012 23:50	Checkpoint 2	closed
Refactoring Code/Changes to Structure?	Should we consider refactoring any of the interrupt code into some other file (such as "interrupt.c")? I think it would help to keep things modular (and prevent kernel.c from becoming horribly long/messy).	3/13/2012 23:35	4/6/2012 14:03	Checkpoint 2	closed

ECE 391 BUG LOG

TEAM XI

RTC does not fire	RTC is init'ed and IRQ 8 enabled but interrupt never called	3/12/2012 17:17	4/30/2012 16:26	Checkpoint 1	closed
Keyboard interrupt only fires once	first keyboard interrupt fires, future ones do not	3/12/2012 17:15	4/30/2012 16:25	Checkpoint 1	closed
Initialize paging	Create page directory and a page table.	3/7/2012 22:18	3/12/2012 1:37	Checkpoint 1	closed
Initialize the devices	For now, every interrupt should call `test_interrupts`.	3/7/2012 22:16	3/12/2012 18:02	Checkpoint 1	closed
Initialize the IDT	First init IDTR with the LIDT instruction	3/7/2012 22:15	3/12/2012 0:32	Checkpoint 1	closed
Load the GDT	init CR0 through CR4, then set PE flag (bit 0 of CR0) to start the proc vol 3, 9.8.1: something involving the LGDT instruction.	3/7/2012 22:11	3/12/2012 0:32	Checkpoint 1	closed
Read the spec		3/7/2012 18:14	3/12/2012 0:32	Checkpoint 1	closed