

Checkpoint 1 Bug Log

Name	Type	Symptom	Find Difficulty	Fix Difficulty
Warning for incorrect ptr type for room name	Semantic error	"warning: passing argument 2 of 'draw_statusbar' from incompatible pointer type"	Easy (adventure.c)	Realized that I should not have been passing the ptr to the room object, but rather the actual name of the room (function room_name found in world.c)
Warning for char subscript to array	Semantic error	"warning: array subscript has type 'char'"	Easy (text.c)	Force (int) type for subscript to font_data array
Fix target_img in set_mode_X()	Algorithmic error	Rapid flashing on screen between two status bars	Very difficult (had to consult others or I would have never gotten this) (modex.c)	Simply fix start offset of video memory to 0x1680, the size of our status bar
Fix CRTC ModeX register values	Algorithmic error	Two status bars showing, one rendering in the middle of the screen	Easy (modex.c)	Added the high bit back to line compare field, originally thought that line compare should be $200 - 18 = 182$, but must be doubled for 2 scan lines/px
Switch command executes entire set of statements	Semantic error	Text always stuck at right position despite print_start and position variables (typed command prints over room name)	Moderate (text.c)	Switched position var from int to char type (for clarity); added 'break' after each statement

Warning for copy_statusbar not used	Semantic error	“Warning: copy_statusbar defined but not used”	Easy (Modex.c)	Damn near impossible. I ended up just having to remove the static qualifier to get rid of the warning
-------------------------------------	----------------	--	----------------	---

Checkpoint 2

Name	Type	Symptom	Find Difficulty	Fix Difficulty
Was not using copy_to_user in TUX_BUTTONS but tried to directly set pointer	Algorithmic error	Button status not being updated	Moderate (ioctl.c)	Utilized copy_to_user in ioctl_buttons
Segfault when setting display time value	Semantic error	“SEGFault in display_time_on_tux”	Easy (input.c)	Had to declare using array with fixed value 4 (accidentally declared using i)
Introduce local_ack to handle reset commands to LEDs	Algorithmic error	LEDs reset to blank and controller unresponsive even though they were supposed to reset to previous val	Moderate(input.c)	Had to introduce local_ack and check before setting LEDs to ensure previous LED set complete
Breaks missing from tuxctl-ioctl	Semantic error	TUX_BUTTONS executes without button press	Moderate (ioctl.c)	Add breaks
Mixed declarations and code warning	Semantic error	Warning for mixed declarations and code	Easy (ioctl.c)	Moved all declarations together at top of function
Copy_to_user	Semantic error	Button status still	Moderate	I was essentially

ptr failure		not being updated	(ioctl.c)	accidentally creating a double ptr by passing a ptr type into the (void*) argument of copy_to_user
Warning about declaring unsigned long flags inside switch case	Semantic error	Strange warning; could not figure out cause	Moderate (ioctl.c)	Created helper function called on mtcp_bioc_event opcode
No longer resetting curr_leds in init()	Algorithmic error	LEDs still clearing on reset	Moderate (ioctl.c)	Remove statement resetting curr_leds to zero on init()
Sending nonsensical empty packets because buffer size was fixed to 6 even if we only wanted less than four LEDs lit	Algorithmic error	LEDs would not update, or randomly update to incorrect sequences	Moderate (ioctl.c)	Introduce buf_len var to keep track of how long the LED buffer sent to tux should be
Poll_driver function had to return unsigned long	Algorithmic error	Button status stuck at 0xFF in adventure.c	Moderate (input.c)	Change poll_driver() to return unsigned long instead of void
Add additional pop_cleanup(1); move status thread to after init_input()	Semantic error	Error for not reaching end of function execution	Moderate (adventure.c)	Add pop_cleanup(1) after tux thread creation in adventure.c
Introduce terminate var to close game	Semantic error	Could not return non-ptr from tux_thread	Easy (adventure.c)	Introduce terminate variable to recognize if cmd_quit should have been returned