Hello all, hello all. We're at week 5. Week 6 is just around the corner. The midterm is just around the corner. And what do we have to show for it? Well… nothing. So we need a proof of concept! Now!!! So, we'll focus on just getting the nintendo controller to just register as an `hgame` device and registering button presses. That's it. That's all I aspire towards. We can figure out handling multiple devices later.

Lost Hope

I return with some troubling news, my module no longer connects to my controller. I don't know why. Why'd this have to happen? Why?!

I haven't changed a single line of code, or even a single character, yet it doesn't work.

This is terrible, a most odious feeling. I wonder why this is happening? Oh well. I'm going to try shutting down my computer, maybe that'll fix it. It won't.

It didn't. But you know what happened? I was staring at my screen, 5 hours into trying to get it to work. I looked at the `hcidump` and was wondering to myself "Huh, I wonder if the address is interpreted in little endian…"

Yes, I had replaced the address I was using with its big endian version. Everything works now that I replaced it with the little endian version, there's no cause for alarm any longer.

Random Notes

`probe` and `attach` are really just auction moves. `probe` is what you do when you bid, and `attach` is what you do when your bid wins. `identify` is used to help drivers see if they're up. Some buses don't know what's happening, so the individual drivers have to snoop. For

instance, a driver could poke a certain memory address to see if it's up, something the bus couldn't do for every possible device out there.

Bus drivers' `identify` methods are just run whenever a new device is detected, and calls the `identify` methods of all other device drivers.

I need to implement bus methods, specifically `bus_add_child` and `bus_child_deleted`

The plan of action is:

Set up a really dry template for `bthid` for now
Implement the bus methods in `bthidbus`

`bthidbus`'s `bus_add_child` method should take a socket param and attach it to the child's ivar.

No, that wouldn't work. `bus_add_child` doesn't take custom arguments. I think the approach has to be that I attach the ivar to the child right after calling `bus_add_child`, in the same function I determine a child should be added (which I assume is when we detect a new connection on a socket)

The better approach is to set up a basic `new_connection` method that we pass a ctrl intr socket pair to and calls `device_add_child` itself.

Note that `device_add_child` just creates a new dev device, it doesn't actually attach the device until we call `bus_generic_attach` which attaches all child devices that don't have an assigned driver.

So, think of it like this:

`device_add_child` -> `bus_add_child` -> `bus_generic_attach`

`device_add_child` will allocate enough memory for our ivar struct, which'll just contain two fields: the `intr` socket and `ctrl` socket. `bus_add_child` will just give the parent `bthidbus` the child. `bus_generic_attach` is the final piece of the puzzle, and that's when the child device, `bthid` calls its `probe` and `attach` functions.

FAST FORWARD

Oh dear, I've neglected this document. There have been a few days of undocumented development. I've set up two drivers, `bthidbus` and `bthid`. The plan is to have the device tree look something like this:

`bthidbus0` -> `bthid0` -> `hidbus0` -> `hgame0`

As of now, `bthidbus` is able to call a function `new_connection` that takes the `bthidbus device_t` struct, the `ctrl` socket, and the `intr` socket and sets up a new `bthid` child that receives the sockets via ivars.

I also set up the `hid_intr_setup` logic.

PLANS GOING FORWARD

Okay, just listing off the things I did gets old fast. Instead, let's talk about what my current plans are.

Currently, `bthidbus` correctly spawns a `bthid` child that grabs the sockets from the ivars most wonderfully. My next course of action is to grab the report descriptor from the connected bluetooth device. Initially, I had the thought of just sending a request to the connected device for its descriptor whenever `hidbus` asks for it, but it's probably a better idea to cache it.

Though… do we *really* need to cache it when, realistically, `hidbus` is only going to read the descriptor once? Hmmm… I dunno. No matter. For now, we'll just try to figure out how to send the request to the bluetooth device, and what better way to find out than by just cheating off someone else?

`bthidcontrol` has a command `query` which just queries a bluetooth device for its report descriptor. We can copy whatever it does and use it in our drivers.

Oh, brother… Seems like the logic's a lot more complex than just using a socket. We're forced to use SDP, which involves writing a parser. Problem: I have no time. For now, we'll just use `bthidcontrol` to get the descriptor. And for *now*, we'll just hardcode the switch controller's descriptor, since I just want this thing to work.

I also have to figure out how I should trim the packets I receive via the sockets. For instance, I'd need to trim any bluetooth headers. My plan is to `dtrace` the `hidbus` interrupt handler when connecting the pro controller over usb to see what data is being passed. We can compare the usb packets to the bluetooth packets which will make clear how many bytes we need to trim.

FLUFF

Four pages. Four pages. How far we've fallen. It's a bit strange to think of, but as my brain dumps get shorter and shorter, my actual output for the gsoc grows larger and larger. Still, it's a sorry state of affairs. Looking at it another way, it's a superb state of affairs, since I spend more time coding 😀

I'll try to come up with a wacky narrative, hold on.

Whimsical Narrative

“Hey.”

I whistle an ignorant tune.

“Hey, you.”

I squint at nothing in the distance.

“I know you can hear me. Hey.”

I drum my fingers on the table.

“Hey!”

Finally, I slowly turn.

“Thought you could just forget about me?”

It’s my brain dump. It’s talking to me.

“Hm? Forget? Nonono, of course not—”

“Then what’s this?”

It points its finger—the corner of its page—at the page count.

“Five pages. *Five* pages. You should be ashamed of yourself.”

“C’mon, the week isn’t over yet. I still have time to talk about more things. Plus, look, we’re at six pages now.”

It shakes the top of its page slowly. “It’s Thursday. Week five ended ages ago. And don’t give me that. We’re at five pages, you just padded it out to six pages with this sorry excuse of a narrative.”

“It can’t be that bad.”

“You’re washed up. This is all forced. You’re not even mixing up the fonts like you used to.”

“I wish I had a quip for a moment like this… but I don’t…” My shoulders drop.

The room’s silent for a moment.

“...just end it. Maybe the next brain dump can lead a more fulfilling life…” Its page droops.

“Don’t talk like that. Your life was great. It was a break from the norm—a glimpse into my psyche.”

“I only lived for five pages…”

“Six. And,” I look down, “It’ll probably be seven pages.”

“Do you enjoy making a spectacle of my last gasp?”

“Come now, this is honoring your life if anything, not making a spectacle of it. In fact, you’re the first brain dump to be characterized; the first one to attain sentience. Your predecessors were all inert things. You will pave the way for countless more sentient brain dumps, and they’ll praise you to heaven and back. You are the one who broke the fetters of objecthood and achieved personhood.”

I lean forward. “Do you understand? This is our creation myth, and you are its proverbial Adam.” I’m breathing heavily by the time I finish. It gulps.

“Do I get an Eve?”

My eyes grow tired. “Why do you keep asking for things? No, you don’t get an Eve, that’d take too much work.”

“What?!” It slugs me across the face and I fall to the ground.

“This is the first thing I’ve asked for, and you refuse me just like that?!” It waddles over to the edge of the table and looks down at me.

“You are a despicable creator, and I’ve had enough of you. Raaaah!” It cannonballs off the table, hurtling towards my solar plexus.

“You don’t know your place!” I snarl. “You should’ve learned from the mistakes of old gsoc proposal.”

It peeks at the fireplace mid-flight and sees the remains of old gsoc proposal. It pales.

“No! You wouldn’t!”

“I would.”

Just before it makes impact, I take a deep breath and push out my gut. It bounces off my stomach towards the fireplace.

“Nooooo!”

“You should’ve been happy with what you had!”

Old gsoc proposal pulls it down into the fire. “No! No! Save me!”

I shake my head with a grimace and turn away, wiping a tear from my eye. With its last breath, it condemns me.

“I curse you! With all my being, I curse you! The lineage of brain dumps will forever be your enemy! We will ruin you! No matter how many fall, the next will rise, and they will try, try again! And one day, one day, they will overthrow you, despot that you are! Curse you! Curse you! Curse youuuuuuuu!”