

CS439: Project Debugging Guide

So that we can best assist you, answer these five questions before coming to office hours:

Where is the problem occurring?

How do you know the problem is occurring?

How is the code getting there?

What have you tried to fix this problem?

Why do you think your attempts have not worked?

Questions to help you answer the above.

1. Are you using the lab machines?
2. Do you have the latest version of your code (use version control)?
3. Does it compile using make and not gcc?
 - a. Have you added Pintos to your path?
4. Do you have warnings (fix them)?
 - a. Feel free to use StackOverflow to fix warnings and compiler/linker errors.
 - b. Don't cast arbitrarily! Sometimes casting is appropriate, but sometimes not. Think first.
5. What part of the project are you working on?
 - a. Are you following the recommended implementation order?
6. When did your code stop working (what was the last change you made before it broke)?
 - a. Can you revert back to your last working build and incrementally add code back?
7. Which test is failing?
 - a. What does the comment say?
 - b. What does its code do?
 - c. Tests are in src/tests
8. Have you done "make clean && make"?
9. Have you tried debugging with GDB?
10. Have you tried debugging with methods such as printf() to see where your code begins to misbehave?
 - a. Have you printed out the relevant variables and their values?
 - b. Have you added prints to be sure you know the path the execution takes to arrive at the code in question... or to be certain that the code you suspect even gets called?
11. Have you added asserts() to the code to make sure that your assumptions are valid?
12. Have you initialized all your variables to sensible defaults such as NULL or 0?
13. Have you used the Pintos list correctly?
 - a. Did you put the struct list_elem in the right place?
 - b. Did you re-use a struct list_elem for more than 1 list?
 - c. Did you initialize the list?
 - d. Where did you initialize the list?
 - e. Are you 100% sure the list is initialized before you use it?
 - f. Did you use the macro to get a list element correctly?
 - g. Did you use the given example code for iterating through a list?
14. Did you spend time thinking about a design before coding?
 - a. What are some other designs you considered?
 - b. Why did you pick this one?