How to deal with context (i.e. pointer declaration vs dereference)

`mips-linux-gnu-gcc -fverbose-asm -S myprog.c` - compile into mips

Big man simon:

Based on some of the questions I got for the compilers coursework today, just wanted to say a couple general things:

- ANSI C Grammar: USE IT. Take the time to read through it and try and thing what kind of AST it builds, start with really simple programs and go from there. There is some stuff in there that isn't relevant / won't make sense yet, you can comment it out / delete it and add it back later.

- Use the lab 2 approach, have a generic 'ast node' type, which all other nodes inherit from, and implement the specific functionality for each node type.

- Godbolt: super helpful tool, if you're not super confident on how the MIPS ISA works check the assembly it produces for different test cases

- Testing: in your testing folder, you will have 2 files per test, test.c and test\_driver.c. Your compiler is used to compile test.c, and it is linked to test\_driver.c which is compiled by gcc (so you only need to worry about making test.c compile, but it needs to be compatible with the code gcc produces). Familiarise yourself with the different test cases given to you, they can help guide what features you implement.

- Start small, try and get default/test\_RETURN.c building a correct AST first, then generate code for it. Once that works start adding more functionality. A good base will make it so much easier to get more advanced features working later down the line.

- Expectations: You can sink infinite time into this coursework. No one in the history of this coursework has ever gotten full marks in it (I think the top mark in my year was 82%). It will take a significant number of hours to get your first test case passing, and it'll be very overwhelming at first, but if you put in the effort you will get things working and it'll start to make more sense. From a pure grades optimization point of view, it's better to get all your other cw done first, they're easier and worth more. From a learning perspective (opinion), this was the coolest coursework I've done at Imperial so far, I learnt so much from it and it was a really good challenge / bonding experience with my partner. It's also a lot more fun that writing lab reports :)

Good luck guys ☺️