

Forum:[Hack Forums](#)

Thread: [Practical applications of MIPS32](#)

6577657534735 - March 2, 2014, 12:52 pm

We're learning MIPS32 in a class and I really like it. One assignment we have to do is making either an assembler, interpreter, or disassembler. I'm planning to make a sort of all in one package since it seems like if you have the framework set for one of them, the other two will be trivial to add on.

But I'm wondering if there are any actual practical applications of this? I don't know of any hardware that uses MIPS architecture except Playstations. How hard would it be to turn a MIPS disassembler/interpreter into a Playstation emulator?

Feminist - March 2, 2014, 5:52 pm

(Mar 2, 2014 01:52 PM)Abraluna Wrote: [\[CITE\]](#) We're learning MIPS32 in a class and I really like it. One assignment we have to do is making either an assembler, interpreter, or disassembler. I'm planning to make a sort of all in one package since it seems like if you have the framework set for one of them, the other two will be trivial to add on.

But I'm wondering if there are any actual practical applications of this? I don't know of any hardware that uses MIPS architecture except Playstations. How hard would it be to turn a MIPS disassembler/interpreter into a Playstation emulator?

It would be incredibly hard to emulate Playstation. I wouldn't even bother trying to do that at all.

The practical application of making an assembler would be so that other people can code in a high level aspect of assembler.

0x00000000 - March 2, 2014, 11:04 pm

Your going to need to know the instruction set, (opcodes and arguments), and an underlying knowledge of the hardware itself. You will probably have to worry about timing and execution.. So it really depends on implementation (like if you are dynamically reading the instructions, or interpreting them in a switch?). I think it would be interesting to make an assembler tbh.

6577657534735 - March 2, 2014, 11:46 pm

(Mar 3, 2014 12:04 AM)0x000000 Wrote: [\[CITE\]](#) Your going to need to know the instruction set, (opcodes and arguments), and an underlying knowledge of the hardware itself. You will probably have to worry about timing and execution.. So it really depends on implementation (like if you are dynamically reading the instructions, or interpreting them in a switch?). I think it would be interesting to make an assembler tbh.

Yeah I'm thinking all the controller interrupts and shit will be a pain now that I'm reading up. Same with rendering video and shit. Definitely more complex than emulating some simple command line program thats for sure.

Think Instead of PSX emulation I'll just stick with plain old asm->machine code and machine code->asm and execution of machine code in terminal. Will be easier but still cool