

COMP1521

WEEK 2 - MIPS

Siyu (Annie) Qiu



Announcements

- Lab start this week!
 - Lab1 has been released, due on Week3 Monday midday
 - Lab2 has been released, due on Week3 Monday midday
- No weekly quiz this week!
- Feel free to email me!

Table of content

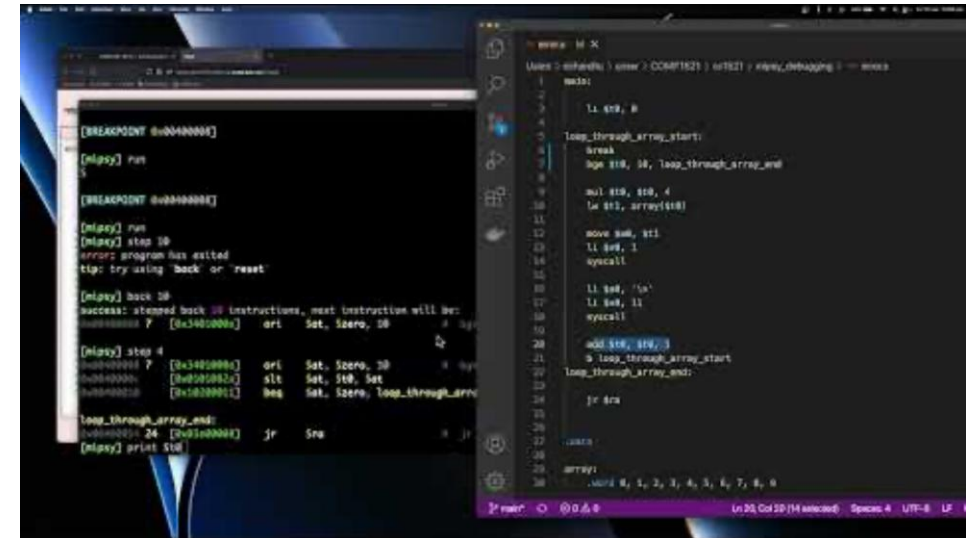
- 01. MIPSY AND MIPSY WEB**
- 02. QUICK REVISION**
- 03. CODING IN MIPS**

Debugging methods

2. Often when writing large MIPS programs, you will make accidental errors that cause your program to misbehave.

Discuss what tools are available to help debug broken MIPS code.

- mipsy-interactive (1521 mipsy)
- mipsy-web (<https://cs1521.web.cse.unsw.edu.au/mipsy/>)
- Line-by-line debugging (stepping)
- Breakpoints
- Printf (syscall) debugging



how to debug in mips:

<https://www.youtube.com/watch?v=25lGn6WNC5o>

(optional): Transform into simplified C first.

- ☐ Recommended by the course
- ☐ Depends on personal taste
- ☐ More useful as you are first starting MIPS.

(optional): Transform into simplified C first.

- ❑ Opposite condition, go to the end
- ❑ Make labels at each condition

```
#include <stdio.h>

#define SQUARE_MAX 46340

int main(void) {
    int x, y;

    printf("Enter a number: ");
    scanf("%d", &x);

    if (x > SQUARE_MAX) {
        printf("square too big for 32 bits\n");
    } else {
        y = x * x;
        printf("%d\n", y);
    }

    return 0;
}
```

```
#include <stdio.h>

#define SQUARE_MAX 46340

int main(void) {
    int x, y;

    printf("Enter a number: ");
    scanf("%d", &x);

    if (x <= SQUARE_MAX) goto x_le_square_max;

    // This is the "else" part of the if-statement.
    printf("square too big for 32 bits\n");

    goto epilogue;

x_le_square_max:
    // This is the "if-then" part of the if-statement.
    y = x * x;
    printf("%d\n", y);

epilogue:
    return 0;
}
```

Environment Setup

- ☐ Try this demo on your laptops!
- ☐ Open MIPS documentation from course website
- ☐ Setup C code and MIPS code side by side.
- ☐ Begin MIPS code with `.text` and `.data` sections
 - ☐ set tab size to spaces: 8
- ☐ Install “Mipsy Editor Features” By Xavier Cooney

Begin Coding! – tutorial q4

- ☐ Turn Brain Off
- ☐ Begin translating line by line.
- ☐ **#Includes** - Nothing to be done.
- ☐ **#defines** - Repeat at the top of .text section.
- ☐ Make a label with the same name
- ☐ Assign them (in comments) to a register.
- ☐ All **String Literals** must go into the **.data**

Run your file

- ☐ **1521 mipsy <name_of_file>**
- ☐ **Or copy code onto MIPSY web to debug!**

Loops

- ❑ A Conditional (if) statement that loops back to itself at the end!
- ❑ Same as conditional statements, take the opposite condition and branch to the end,
- ❑ But create labels for both the end of the conditional statements, and the start!
- ❑ Make sure to loop back to itself.

Tut6, lab question2, tut8

Questions and Answers



3. The MIPS processor has 32 general purpose 32-bit registers, referenced as `$0` .. `$31` . Some of these registers are intended to be used in particular ways by programmers and by the system. For each of the registers below, give their symbolic name and describe their intended use:

Number	Names	Conventional Usage
0	zero	Constant 0
1	at	Reserved for assembler
2,3	v0,v1	Expression evaluation and results of a function
4..7	a0..a3	Arguments 1-4
8..15	t0..t7	Temporary (not preserved across function calls)
16..23	s0..s7	Saved temporary (preserved across function calls)
24,25	t8,t9	Temporary (not preserved across function calls)
26,27	k0,k1	Reserved for Kernel use
28	gp	Global Pointer
29	sp	Stack Pointer
30	fp	Frame Pointer
31	ra	Return Address (used by function call instructions)