**CS 590: Topics in Computer Science**

**Assignment 09: MIPS Debugging and more MIPS GCC**

**MIPS programming**

# Exercise 1: Debugging a MIPS program

Debug the loop in the program in lab09s.s. It is meant to copy integers from memory address

$a0 to memory address $a1, until it reads a zero value. The number of integers copied (up to, but not including the zero value), should be stored in $v0.

Refer: 9-1.mp4

# Exercise 2: Compiling from C to MIPS

The file lab09c.c is a C version of the program in Exercise 1 above. Compile this program into MIPS code using the (cross-compiler) command:

mips-linux-gnu-gcc -S -O2 -fno-delayed-branch lab09c.c -o lab09c\_nodelay.s mips-linux-gnu-gcc -S -O2 lab09c.c -o lab09c\_delay.s

The -O2 option turns on optimization. The -S option generates assembly code.

The above command should generate assembly language output (lab09c\_nodelay.s) for the

C code. Find the assembly code for the loop that copies source values to destination values. Then, for the registers $a0, $a1, $v0, and $v1 from part 2, determine what registers gcc used to store the corresponding value. (For example, $a0 was used to store the source address of integers to be copied. What register is used for this purpose in the mips-gcc output?)

Compare the programs lab09c\_nodelay.s and lab09c\_delay.s, explain what is different and why.

Lab09c\_nodelay.s

$L3:

# Here is the part assembly code for the loop that copies source values to destination values

# $3: dest, $2: source, $5: k

sw $4,0($3)

lw $4,4($2)

addiu $5,$5,1

addiu $3,$3,4

addiu $2,$2,4

bne $4,$0,$L3

nop

#

Lab09c\_nodelay.s

$L3:

# Here is the part assembly code for the loop that copies source values to destination values

# $3: dest, $2: source

sw $4,0($3)

lw $4,4($2)

addiu $5,$5,1

addiu $3,$3,4

bne $4,$0,$L3

addiu $2,$2,4

#

**Questions/Tasks:**

1. Explain what a side-channel attack is.

A side-channel attack is a security exploit that aims to gather information from or influence the program execution of a system by measuring or exploiting indirect effects of the system or its hardware -- rather than targeting the program or its code directly. Most commonly, these attacks aim to exfiltrate sensitive information, including cryptographic keys, by measuring coincidental hardware emissions. A side-channel attack may also be referred to as a sidebar attack or an implementation attack.

1. Explain how an operating system process separation helps in hardware security.
2. Explain how return-oriented programming works to read memory from the victim's process.
3. Explain about the cache timing attack and countermeasures against it.