

COMP1521 Week 4 Lec 1

Angela Finlayson

Announcements

Weekly Quiz 3 Due: Thursday 20:00:00.

Lab 3 Due: Today midday

Census Date : Sunday 10 Mar 2024, 11:59pm

Assignment 1 Due: Week 5 Friday (next week)

Help Sessions [Schedule](#)

Assignment

Assignment:

Watch Video

Fetch Code

Run C Code

Do subset 0 (autotest for subset 0)

Follow style in supplied .s code

Simplified C and autotests simplified C

Overview

Assignment Style

do_while loops

Recap of Functions in MIPS

Integer representation

MIPS Functions

Recap

Basics of function calls in MIPS

jal f

jr \$ra

function_example.s

Inputs: \$a0, \$a1, \$a2, \$a3 int f(int x, int y, int z);

Return Value: \$v0

Register Conventions

Register conventions:

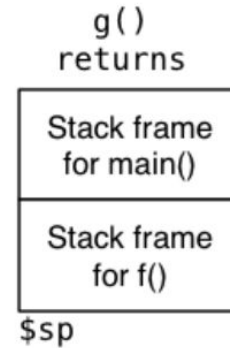
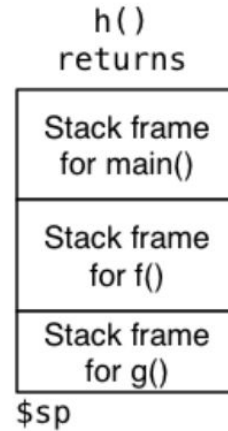
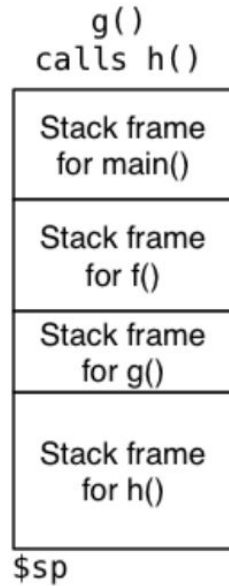
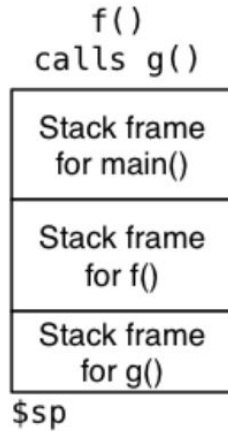
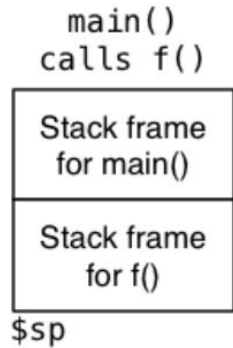
Other function calls may destroy values in : \$a, \$t registers

Functions must not destroy values in \$s registers. (or corrupt \$sp and \$fp).

sum_to.s

sum_to_r.s

Stack Frames



Frame Pointer Easy Mode

- We don't expect you to fully understand the frame pointer in COMP1521
- Instead, we provide you with two pseudo-instructions in mipsy
 - **begin**
 - saves the old \$fp to the stack (keep track of the previous stack frame)
 - sets \$fp to the current \$sp
 - should be the first thing in the prologue
 - **end**
 - restore \$sp to point to the top of the previous stack frame
 - restore the \$fp to point to the previous value of \$fp (bottom of the previous stack frame)
 - should be right before **jr \$ra**
- This makes situations where you push/pop much easier to debug
- Not *necessary* but makes debugging much much easier - **strongly advised**

Function Skeleton

```
func:
    # [header comment]
func__prologue:
    begin
    push    $ra
    push    $s0
    push    $s1

func__body:
    # do stuff

    li     $a0, 42
    jal    foo      # foo(42)

    # foo return val in $v0

func__epilogue:
    pop    $s1
    pop    $s0
    pop    $ra
    end

    jr     $ra
```

Frame Pointer Uses

- Used by debuggers to analyse the stack
- Using a frame pointer is optional (both in COMP1521 and generally)
 - Compilers omit the use of a frame pointer when fast execution/smaller code is a priority
- Used for situations where the stack grows during a functions execution. NOT used in COMP1521.e.g

```
void f(int a) {  
    int length;  
    scanf("%d", &length);  
    int array[length];  
    // ... more code ...  
    printf("%d\n", a);  
}
```

source code for frame_pointer.c

Integers

Integers Motivation

Fundamental topic in computing

Understand what you are seeing in mipsy web!

Understand limits of types and help you understand and debug code

Prepare you for the next topic bitwise operators

Please give Lecture Feedback



If you have any feedback or thoughts about today's lecture, please follow the link below. Please remember to keep your feedback constructive, so I can use it to improve the learning experience.

<https://forms.office.com/r/vxYikTeWQQ>