Procedures

Lecture 10
Section 2.8

Robb T. Koether

Hampden-Sydney College

Mon, Sep 23, 2019

Outline

MIPS Programming Methodology

- Write the program in C.
- Implement the functions.
- In each function
  - Write the function body.
  - Write code to save and restore registers of the caller.

Non-Leaf Functions

- A leaf function is a function that does not call any other function.
- A non-leaf function is a function that does call another function.
- In non-leaf functions, use the `$s` registers for any values that must persist across a function call.
  - The called function will save the ones that it uses.
  - Use `$t` registers for variables that do not need to persist across a function call.
- In leaf functions, use any registers you like.

- Write a function `max()` that will return the value of the larger of two integers.
- Write it once, using only `$t` registers.
- Write it a second time, using `$s` registers.

Outline

Nested Procedures

- When one procedure calls another procedure, we have to be careful.
- The second call (`jal`) will load the newer return address into `$ra`, replacing the previous return address.
- This is disastrous.

Nested Procedures

- To avoid this problem, we have to save the first return address while calling the second function.
- The standard way to do this is to push it onto the stack.
- The stack pointer (register `$sp`) contains the address of the top of the stack.

Saving the Return Address

Save the Return Address

```
    addi    $sp, $sp, -4    # Reserve space for one int
    sw      $ra, 0($sp)     # Store return addr on stack
```

- Push the current return address onto the stack to save it.

Saving the Return Address

Restore the Return Address

```
    lw      $ra, 0($sp)     # Load return addr from stack
    addi    $sp, $sp, 4     # Deallocate the space
```

- Pop the return address off the stack to restore it.

Nested Procedures

Nested Procedures

```
func1:
    addi    $sp,$sp,-4          # Reserve space for one int
    sw      $ra,0($sp)          # Store return addr on stack
      :
    jal     func2               # Call nested procedure
      :
    lw      $ra,0($sp)          # Load return addr from stack
    addi    $sp,$sp,4           # Deallocate the space
    jr      $ra                 # Return
func2:
```

Outline

Finding the Largest of Three Integers

- Write the C code for a program to find the largest of three integers.
- Have the `main()` function call the `max3()` function, which calls the `max()` function.
- Write the program in MIPS.

Outline

Assignment

Assignment

- Read Section 2.8.