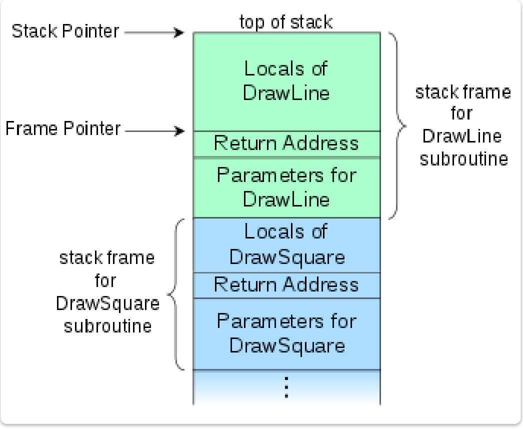
Mips 6

Every time a function/subroutine is called, a stack frame is created for the function

- The stack frame contains areas for
    - Arguments
    - Saved registers
    - Return address
    - Local data
    - Some padding



The function call stack grows downwards

Mips Function call Convention

1. Allocate space in the stack by moving the stack pointer. Calculate the room required by counting the number of registers that require storage 2. Store caller temporary registers (that the caller may use later) - the `t` registers and `a` registers on to the top of the stack 3. Store the `$fp` value onto the stack then copy `$sp` to `$fp` 4. Store the `$ra` value on the stack 5. Save the argument to the `a` registers. If there are move than 4 arguments, store the extra arguments on the stack 6. Call the function using `JAL`

Suppose a function f0 is shown below...

```
funct1: addi $sp, $sp, -4
                sw $ra, 0($sp)
                jal func2
                lw $ra, 0($sp)
                jr $ra
```

- func1 can return to fun0

- func0 cannot return to its caller correctly
- func2 can return to its caller correctly