

Adding two numbers in MIPS

Data section: define some global data that is used by the assembly program to avoid having to define it repeatedly

```
.data
# initialize addresses for data
num1: .word 1
num2: .word 2
result: .word 0
```

Text section: stores program code

Main: entry point for assembly program

```
.text
main:
# load numbers from RAM
lw $t0, num1
lw $t1, num2
# t2 = t0 + t1
add $t2, $t0, $t1
# store result back in RAM
sw $t2, result
```

System call: interact with operating system, ask OS to do something

```
# exit program via syscall
lw $v0, 10
syscall
```

Functions in MIPS

- Can call functions using labels and "jump and link":
 - Label defines address of instruction in memory
 - jal LABEL will store the current PC contents in the special \$ra ("return address") register, and set the PC to the address specified by LABEL
 - jr \$ra will jump back to the address stored in the \$ra register
- Can use stack area of RAM (with the stack pointer register, \$sp) to store data so that it can be restored once the function call has finished