EE2361 - Lecture 9
9/26/16

Reminder - HW 1 due Wed
HW 2 middle of this week, due next Wed (10/5)

Subroutines (Functions, Procedures)

Instructions CALL, RCALL
RETURN

These instructions do more than transfer program execution

When you go and execute a subroutine you need
1. Store the return address.
2. Pass values to the subroutine
3. Return results

How can you pass parameters to the subroutine? (parameters, arguments)

2 possibilities.

1. put the values of the parameters in registers.
2. put the values in memory (problematic)
3. put the values on the stack

XC16 compiler (and C) we typically pass parameter to a subroutine on the stack

Structure of the stack frame

For passing parameters

C
- call by value
  copies of the variables to be passed to the function ~~are put on the stack~~
- call by reference
  pointers to the original variable are passed to the function

Need to also be concerned about the sizes of data which are passed to a function.

- "small" values are usually passed as values in registers or on the stack
- "large" values, arrays for example, are typically passed as pointers

Look at a simple example

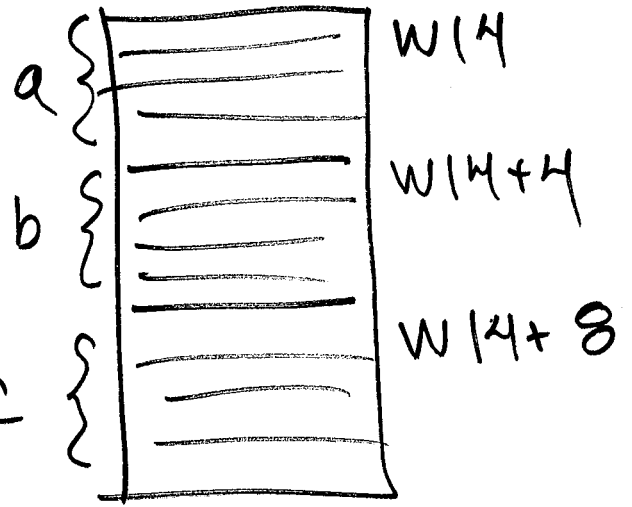
Simple routine that calls a function to add 2 numbers and return the result

```
main routine
  long a, b, c;        // 32-bit integers
    a = 1;
    b = 2;
  c = add2(a, b);
    return c; }
```

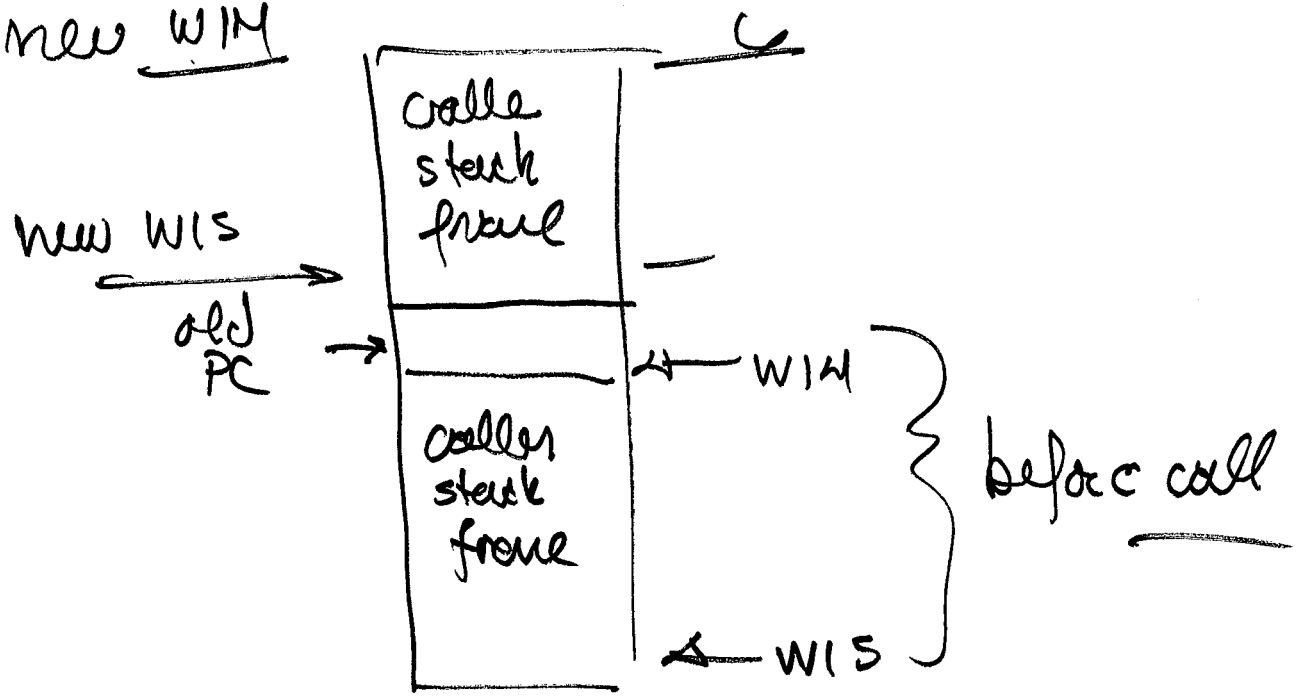
Data types in C —

Example

caller stack



These values appear to be passed in registers


new WIM
new WIS
old PC
caller stack frame
caller stack frame
WIM
WIS
before call