Stacks and Stack Frames

Function Entry

- Want to save the return address ($ra)
- Want to establish a “stack frame”

What’s a Stack Frame?

- Sometimes called a “activation record”, it is a saved copy of the stack pointer at time of entry
- Saved in the “frame pointer” ($fp)
- $fp can be used to quickly restore $sp when exiting a function
- can then restore caller’s $fp and $ra
- Allows functions to be re-entrant/recursive

Notion of “Local Variables”

- Known only for duration of the function
- For Example:

```
int addemup(int a, int b)
    {
    int x, y;    /* Local int variables x and y */
    x = a + b;
    y = x * x;  etc.
    return(y);
    }
```

Local Variables live on the Stack

- On entry:
  - Save return address
  - Save “frame pointer”
  - Copy stack pointer to frame pointer
  - subtract N from stack pointer
  - save registers (push them on stack)
  - possible arguments to another function (beyond $a0 .. $a3)

Frame pointer becomes a chain of “activation records”

Local Variable Access

- Can be accessed as negative offsets from $fp
- For example, on entry:

```
-  addiu $sp, $sp, -4
   sw $ra, 0($sp)  // return address
   addiu $sp, $sp, -4
   sw $fp, 0(sp)
   move $fp, $sp  // copy stack pointer
   addiu $sp, $sp, -8    // make room for x,y
```