CASPER

CS305: Computer Architecture

World of Instructions-VI (The MIPS language)

https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html

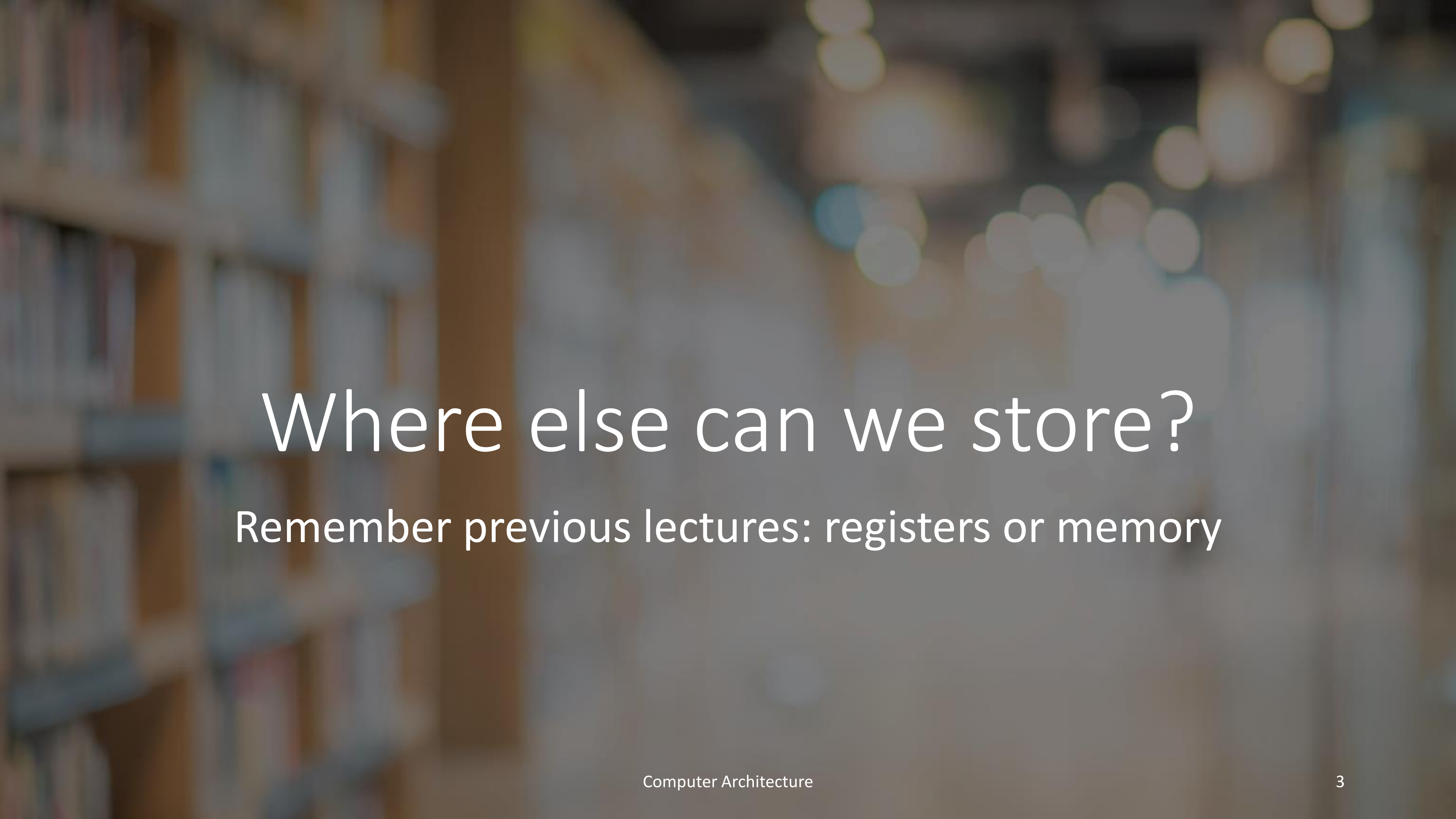
https://www.cse.iitb.ac.in/~biswa/

Quick recap

Register spilling, 32 MIPS registers, nested functions,

oh no!

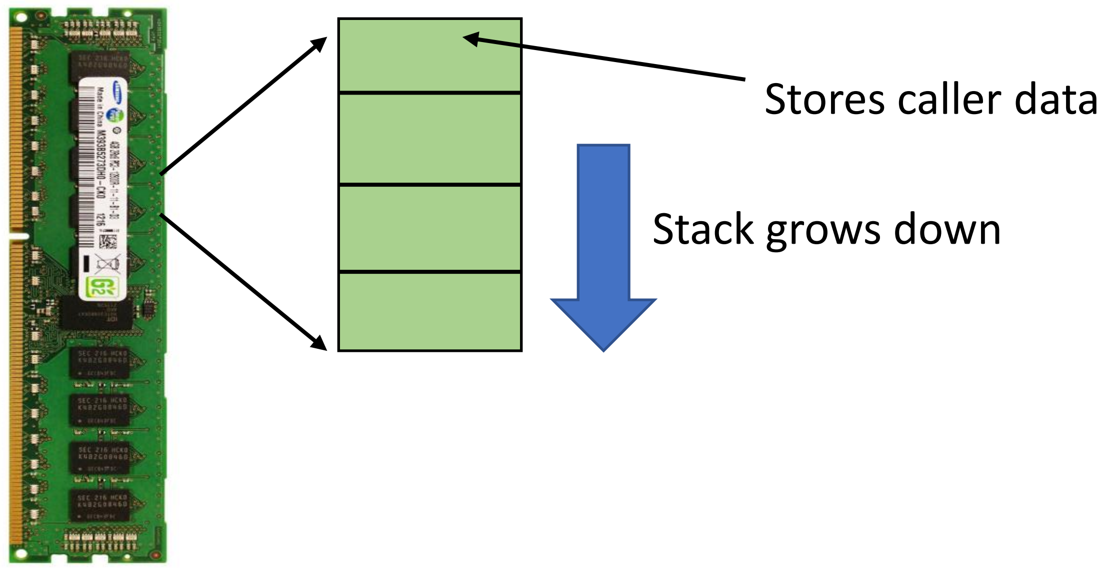
Spilled registers: Where else can we store?



Where else can we store?

Remember previous lectures: registers or memory

MIPS way of handling it: The Stack (part of DRAM, for each function call)



$sp (stack pointer) points to the address where stack ends

One per function, private memory area, else the same problem ☹

Before that: Who does what?

In MIPS,

$t0 to $t9 (R8 to R15, R24, and R25) are temporary and *caller* saved registers. Register values not preserved across function calls (call-clobbered).

$s0 to $s7 (R16 to R23) are *callee* saved registers. Register values are preserved across function calls (call-preserved).

$sp and $ra are caller or callee saved registers ?

Before that: Who does what?

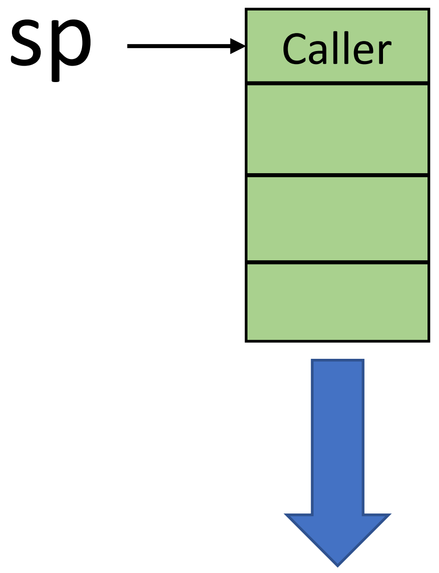
In MIPS,

$t0 to $t9 (R8 to R15, R24, and R25) are temporary and *caller* saved registers. Register values not preserved across function calls (call-clobbered).

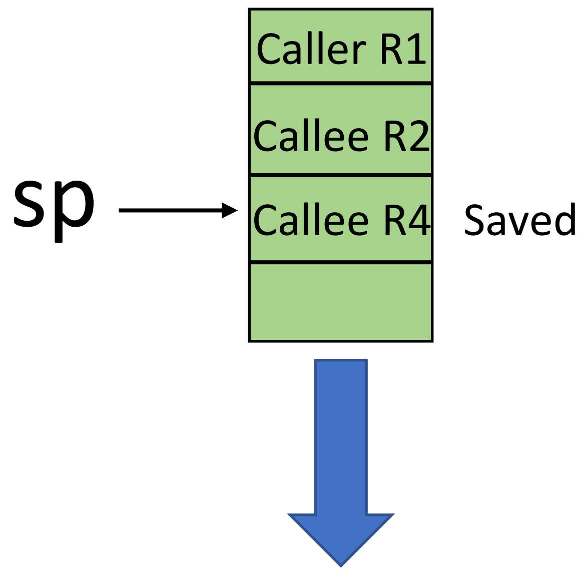
$s0 to $s7 (R16 to R23) are *callee* saved registers. Register values are preserved across function calls (call-preserved).

$sp and $ra are callee saved registers.

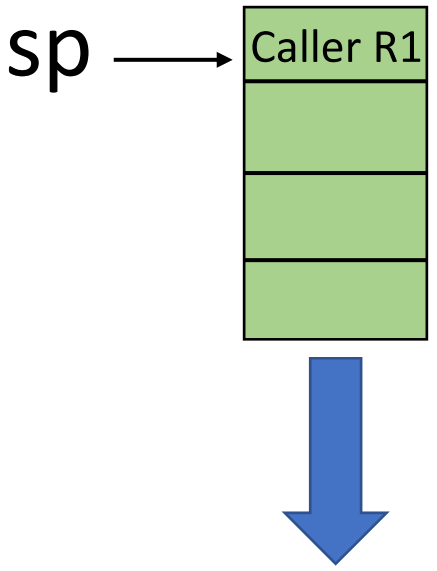
MIPS way of handling it: Before function call



MIPS way of handling it: Function call is ON



MIPS way of handling it: After the function call

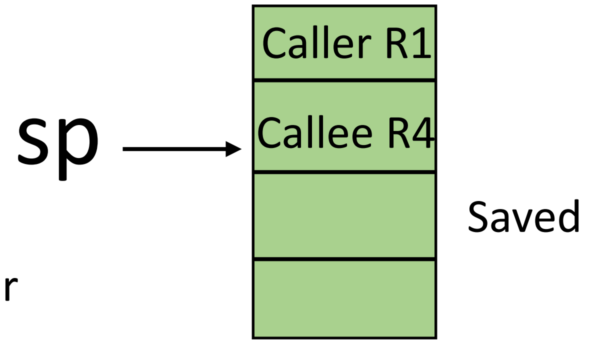


How to save and restore?

Save:

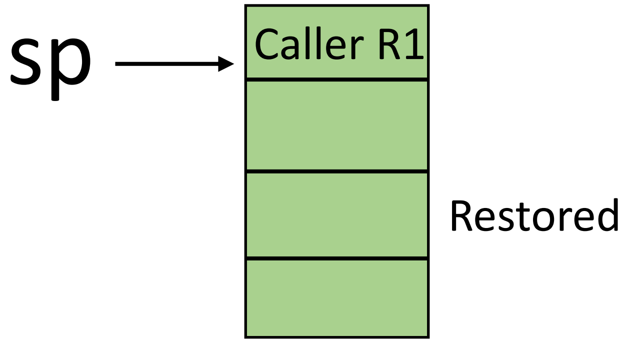
```
addi $sp, $sp, -4
sw R4, ($sp)
```

32 bit registers, 4 bytes, one word, remember



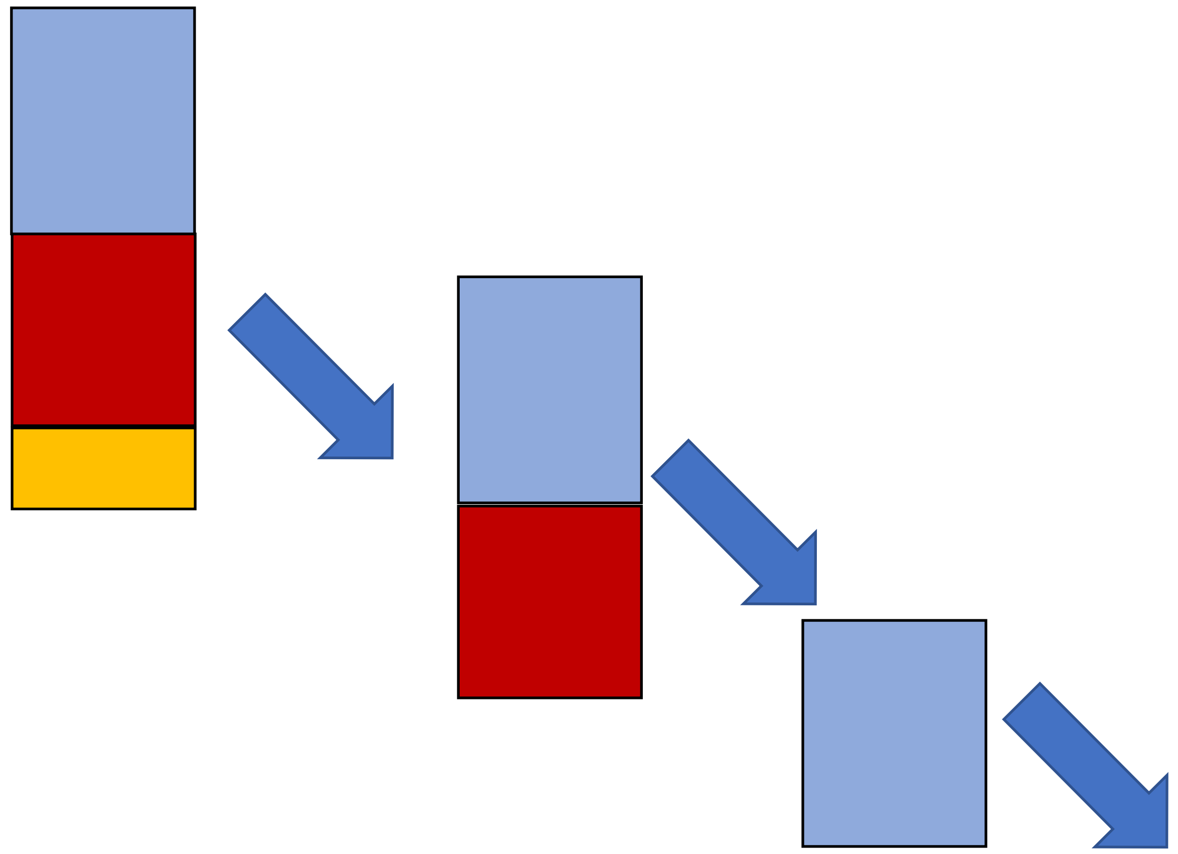
Restore:

```
lw R4, ($sp)
addi $sp, $sp, 4
```



Nested Functions (Remember main() is a function too ☺)

```
CS305 // jal cs305
{
    CS405 // jal cs405
     {
         CS505 // jal cs505
          {
          } //jr
      } //jr
} // jr
```

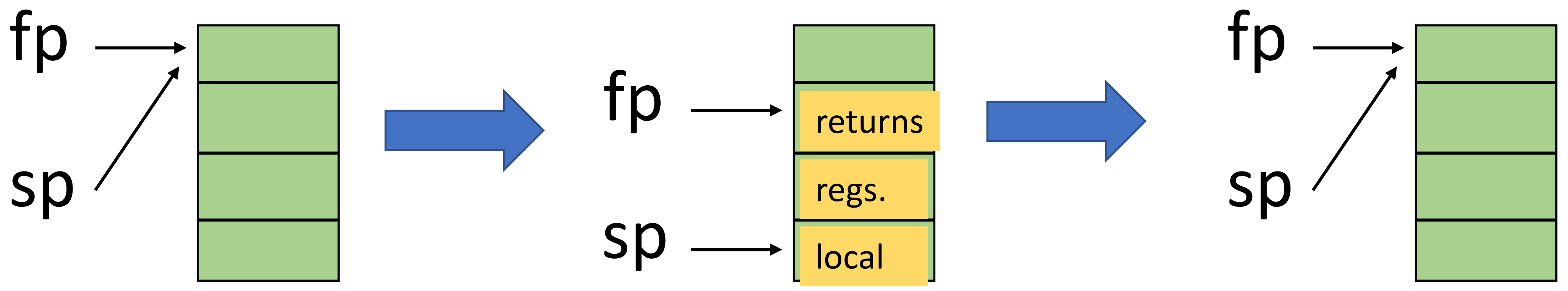


The final one: Frame pointer

Stack also stores local variables and data structures (local arrays and structures) for a function along with the return address(es).

The final one: Frame pointer

Frame pointer: Points to local variables and saved registers. Points to the highest address in the procedure frame. Stays there throughout the procedure. Stack pointer, moves around.



Awesomeness: You can access any using fp/sp and an offset

Try This Out! Discuss on Piazza

Page no A-27 to A-29 P&H

Recursive function fact(n)

Look for sp, fp, ra, jal, and jr

More from the TAs during the labs.

For the Curious Ones (Beyond CS305)

Stack buffer overflow - 101:
https://en.wikipedia.org/wiki/Stack_buffer_overflow

Shukran