



Module 4

MIPS Assembly Language;
Control Structures



Module Four

- MIPS Assembly Language; Control Structures - Part Three
- In this presentation, we are going to talk about :
- Data structure support for Subroutines

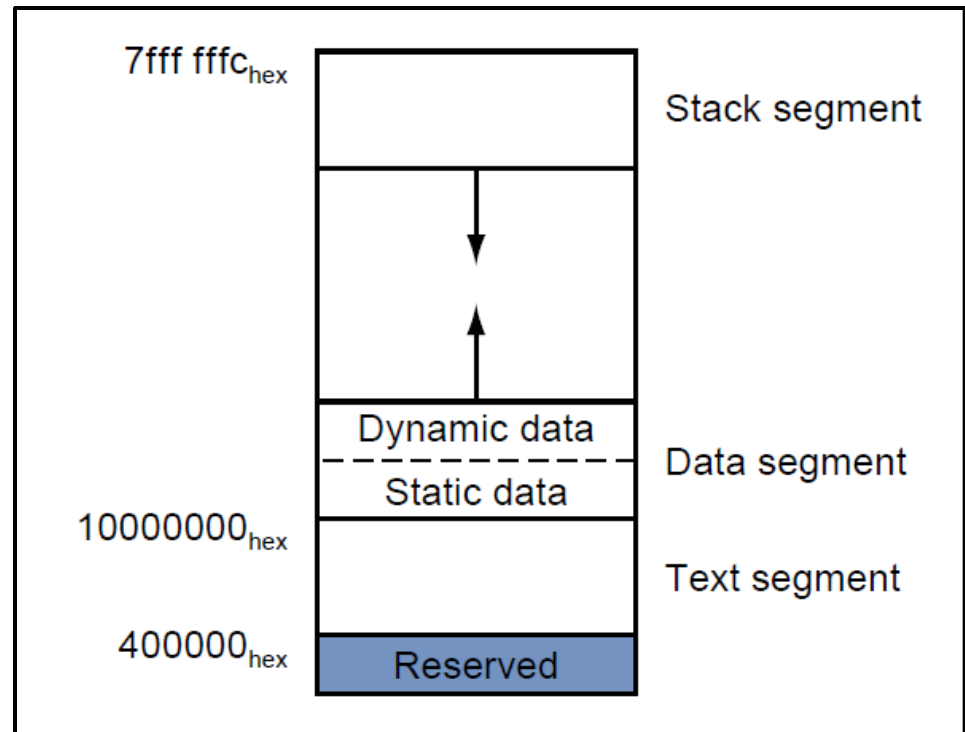


Overview

- Previously we talked about:
 - Process Flow
 - Control Structures
 - Subroutines
- Now: Data structure support for Subroutines

The Memory Stack

- Memory allocation diagram
- The memory stack segment is placed at the high address end of the allocated memory.
- It is allowed to grow.
- The Stack pointer, register **\$sp**, points to the lowest active address on the stack; the TOP of the stack.

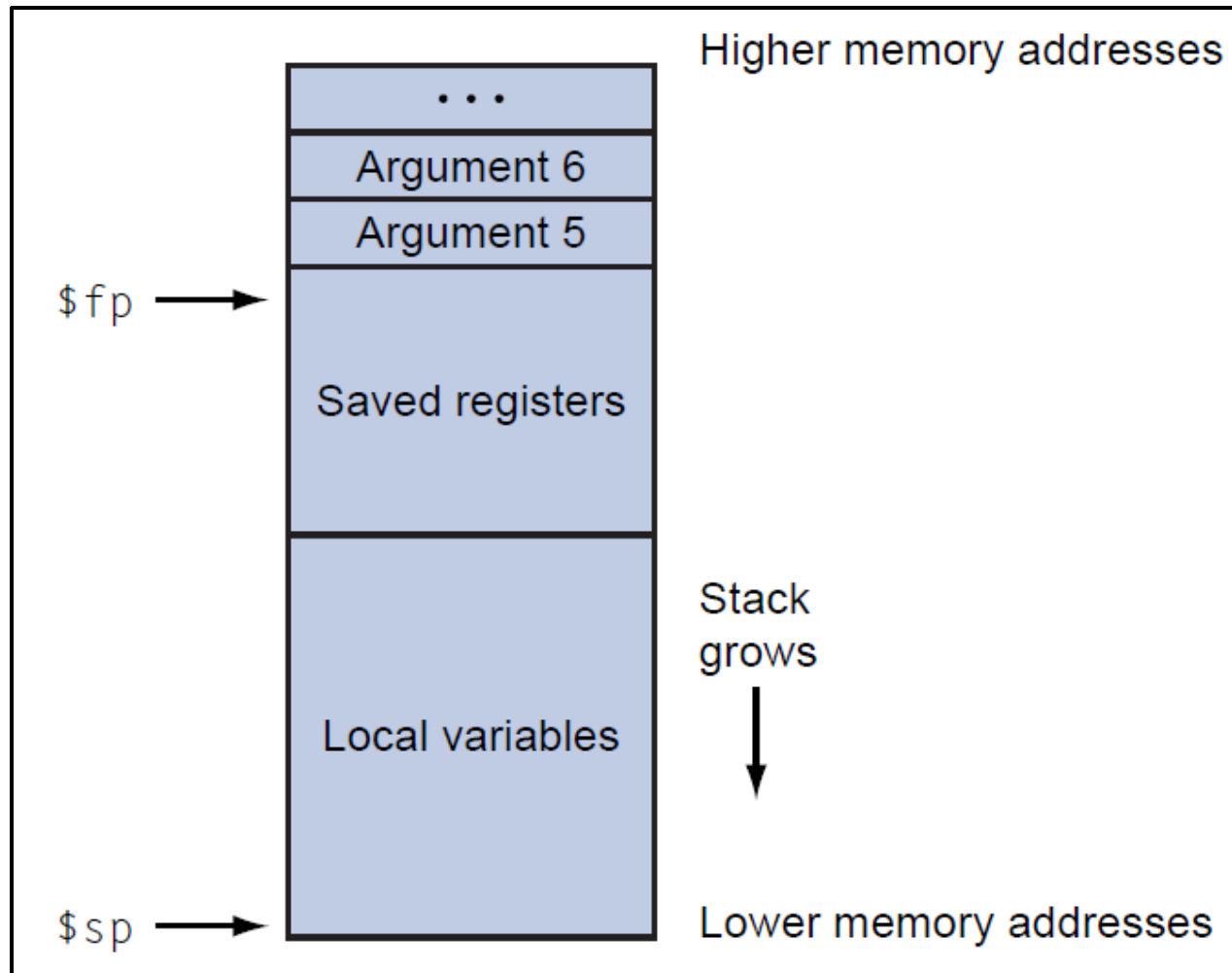




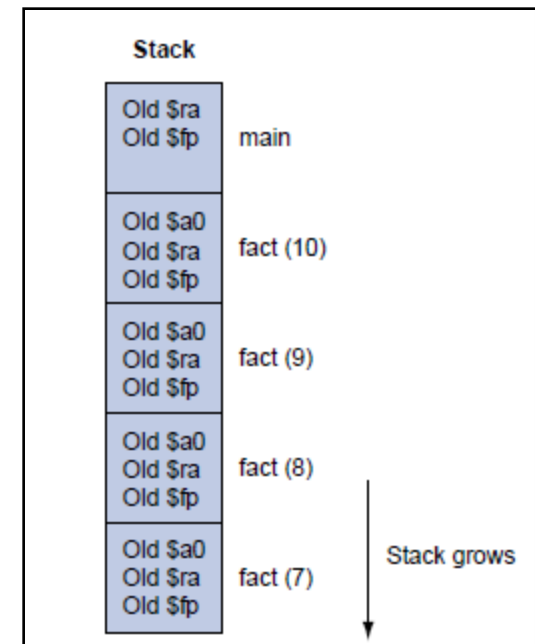
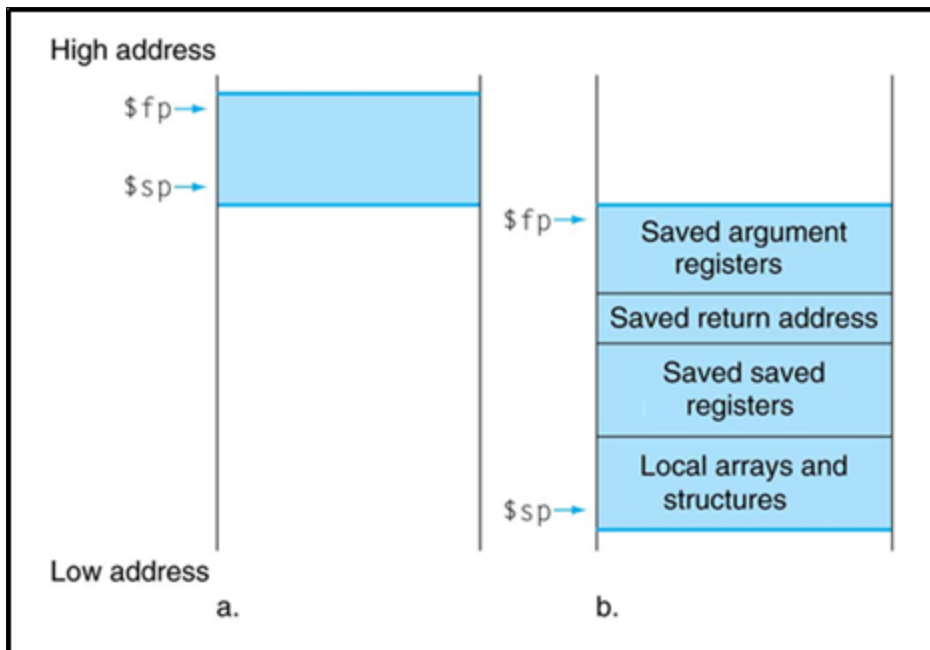
The Memory Stack

- The memory stack is used to store a variable number of values.
- The Stack pointer, register **\$sp**, points to the last placed value on the stack.
- The frame pointer, register **\$fp**, points to the stack location where the subroutine first used the stack to place needed values.

Memory Stack



Memory Allocation





Prolog and Epilog

- Subroutine processing often requires the use of many of the registers.
- Standard sets of instructions are used to save register values and then restore them after the subroutine is complete.
- These are called **prolog code** for the save before instructions, and **epilog code** for the restore after instructions.



Prolog and Epilog

- The **prolog code**, for the save before instructions.

```
ADDIU $sp, $sp, -20
SW     $ra, 16 ($sp)
SW     $s0, 12 ($sp)
SW     $s1,  8 ($sp)
SW     $s2,  4 ($sp)
SW     $s3,  0 ($sp)
```



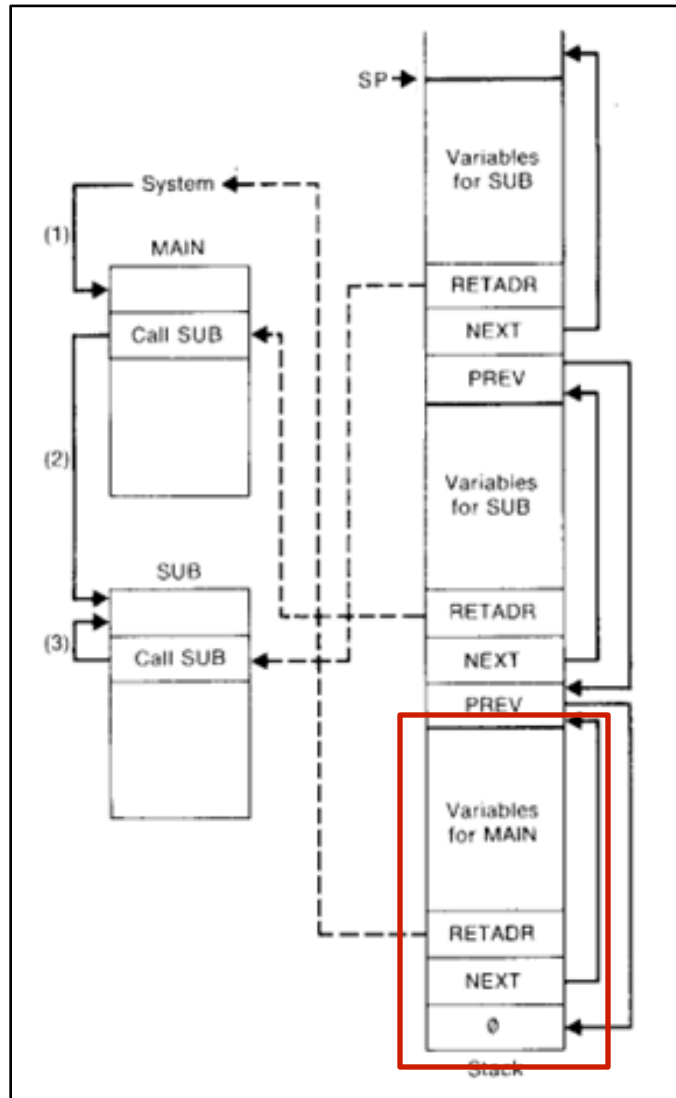
Prolog and Epilog

- The **epilog code**, for the restore after instructions.

```
LW    $ra, 16 ($sp)
LW    $s0, 12 ($sp)
LW    $s1,  8 ($sp)
LW    $s2,  4 ($sp)
LW    $s3,  0 ($sp)
ADDIU $sp, $sp, 20
JR    $ra
```

Block storage allocation

- As each subroutine is called, Prolog code moves the pointers to establish the needed space on the stack
- The addresses are then all relative to the stack pointer.
- NEXT is a pointer to the next free space on the stack; $\$SP - 4$
- PREV points to beginning of this block; $\$FP$





Summary

- MIPS Assembly Language and control structures
 - Process Flow
 - Control Structures
 - IF - THEN
 - Loops
 - Subroutines
 - Data structure support for Subroutines