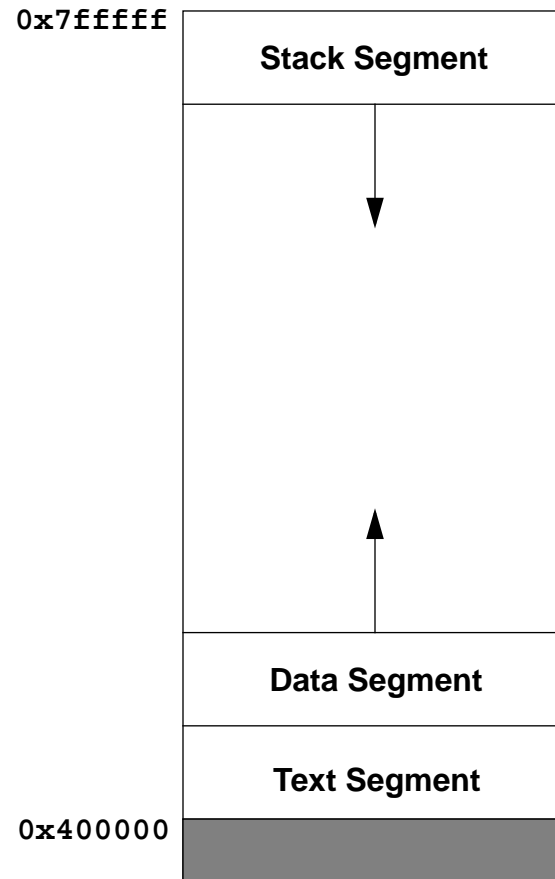
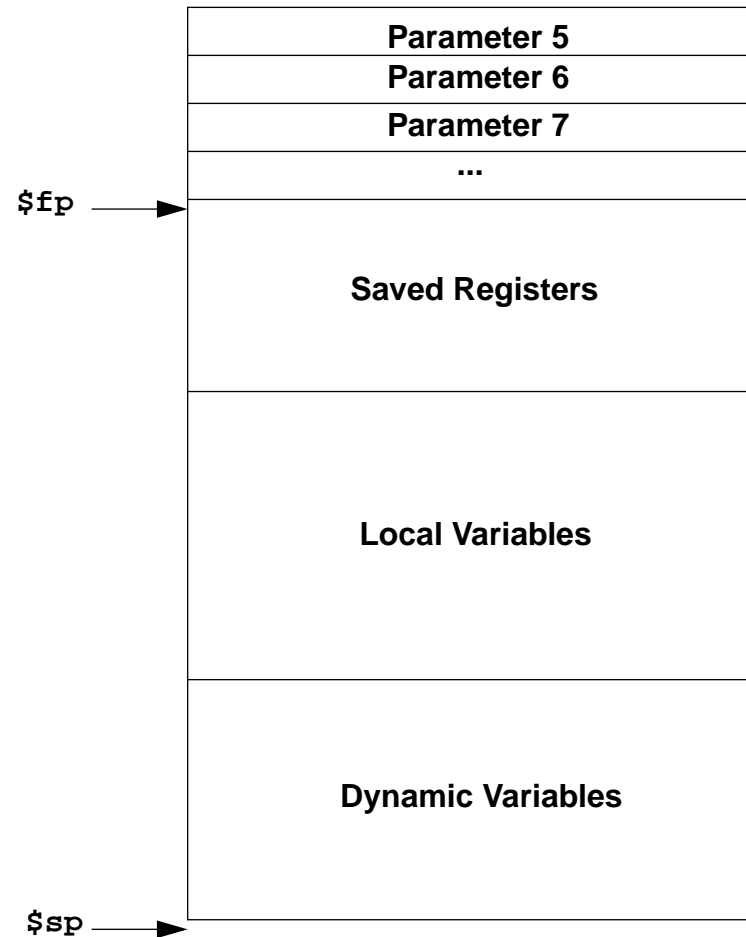


SPIM Memory Layout



Layout of Memory

Layout of a Stack Frame



Procedures: No Parameters, No Recursion

Calling a Procedure:

- Execute a **jal** instruction.

At the beginning of the called Procedure:

- Space is allocated statically for the local variables, so nothing need be done to allocate a stack frame.
- Register **\$ra** needs to be saved in a designated storage location for this procedure in order to allow the routine itself to make calls. Any of the registers **\$s0-\$s7** that are used by the callee need to be saved.

To return from a call:

- Restore any callee-saved registers that were saved upon entry.
- Return by jumping to the return address saved from register **\$ra** using the **jr** instruction

Procedures: No Parameters, with Recursion

Calling a Procedure:

- Execute a **jal** instruction.

At the beginning of the called Procedure:

- Register **\$ra** needs to be saved to the location referenced by the stack pointer
- The appropriate display register must be saved into the following word of the stack frame. Save any other callee-saved registers in the frame.
- Establish the new display pointer by copying **\$sp** into the appropriate display register.
- Establish the stack frame by subtracting the frame size from the stack pointer.

To return from a call:

- Pop the stack frame by adding the frame size to **\$sp**.
- Restore callee-saved registers, including the appropriate display register.
- Return by jumping to the address in register **\$ra** (it is also at the location now referenced by **\$sp**).

Procedures: Parameters and Recursion with Chains

Calling a Procedure:

- Pass the parameters. By convention, the first four parameters are passed in registers **\$a0-\$a3**. The remaining arguments are pushed on the stack
- Set up the static chain register (necessary if the procedure is nested within another dynamically allocated scope).
- Save the caller-saved registers, **\$t0-\$t9**, if they contain live values at the call site.
- Execute a **jal** instruction.

At the beginning of the called Procedure:

- Save the callee-saved registers in the frame. Register **\$fp** must be saved since it is being used to reference variables in the calling frame. Register **\$ra** needs to be saved in order to allow the routine itself to make calls. Any of the registers **\$s0-\$s7** that are used by the callee need to be saved.
- Establish the stack frame by subtracting the frame size from the stack pointer. Establish the frame pointer **\$fp** by adding the stack frame size to **\$sp**.

Procedures: Parameters and Recursion with Chains

To return from a call:

- Restore any callee-saved registers that were saved upon entry.
- Pop the stack frame by adding the frame size + parameter space size to **\$sp**.
- Return by jumping to the address in register **\$ra**.

Upon returning to point of call:

- Restore any caller-saved registers that were saved upon entry.
- Reset **\$sp** by subtracting size of space for caller-saved registers and parameters passed in the stack.