Assembler, Linker, and SPIM

October 10, 2008

Pseudo-instructions

- Instructions for assembler, not implemented by hardware
  - `mov` is a modified `add`
  - `blt` = `slt` + `bne`

Need and Purpose of a Linker

- Separate compilation results in separate `object` files
- Linking: combine separate object modules into an executable
- Resolves external references in all modules

Tasks of a Linker

- Resolves external references among modules
- Searches libraries for required procedures
- Determines memory location for each module and relocates absolute references

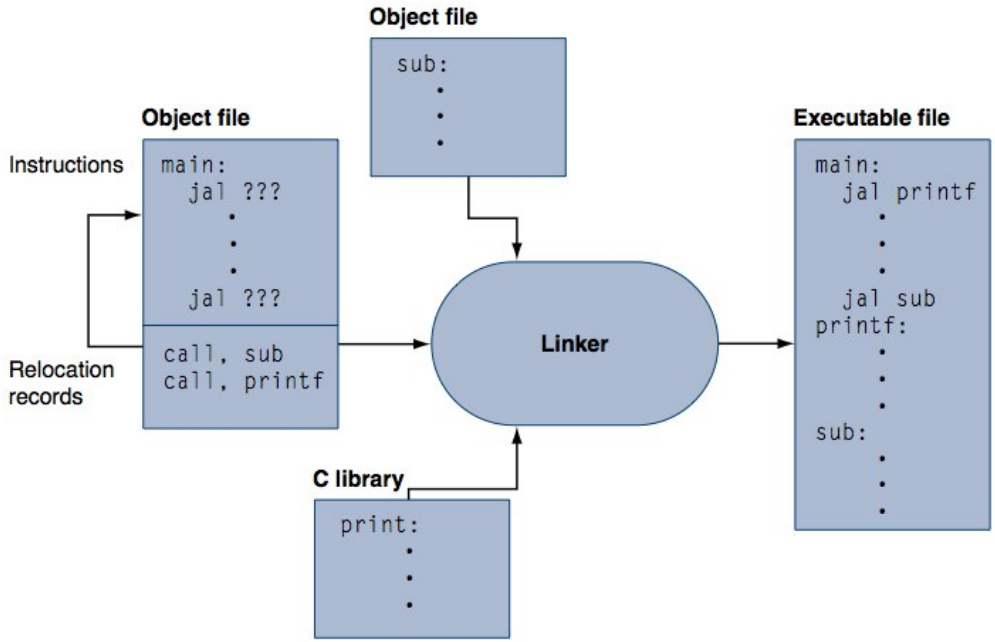

Figure: What Linker does

Loading

- Read executable's header for size of (global) data & header sections
- Creates space in memory for them
- Copies instructions in data to memory
- Copies arguments to `main()` onto stack
- Initializes (clears) registers, including `SP`.
- Jumps to a procedure that
  - copies arguments from stack to `$a0 - $a3`
  - calls `main()` procedure

What is the difference

- All files are binary at storage level
- Binary file's content is analyzed in many different ways
- Plain text files are a subset
  - `.txt` or `ASCII`
  - all source code files
- Text files are read 8 (rather 7) bits at a time
  - Each byte is an ASCII code