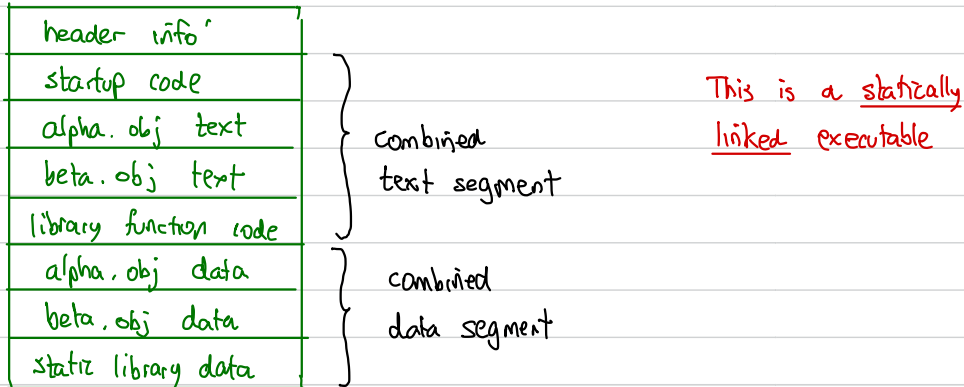


The executable file (continued)

Typical layout for "a.exe" from gcc alpha.c beta.c



Running this file: The OS copies text and data from filesystem to memory; PC set to address of first instruction in the text segment.

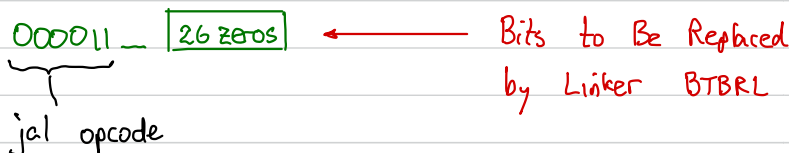
Linking and relocation information in object files

Consider a MIPS-based computer running linux

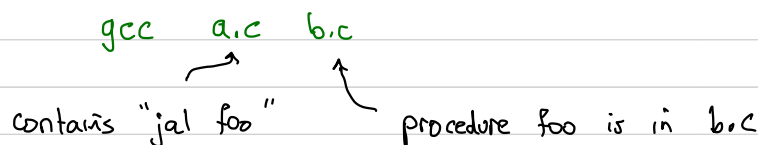
- the assembler can translate most MIPS instruction correctly
- Trouble with some instructions like j, jal, la

e.g. jal foo ← what if foo is in a different file?
(assembler won't know its address)

The assembler produces the code:



Suppose we are compiling:



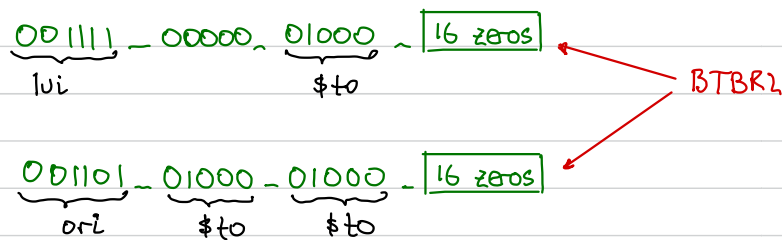
then an entry is made in the relocation information of a.obj.

- tells the linker to fill in the 26 bits when the linker is run

And suppose

lca \$t0 tomatoes ← assume tomatoes is in a different file

Assembler produces two instructions



and makes an entry in the relocation information.

Part of the linker's job

Use relocation info' and symbol table sections of the object files to fill in the blanks left by the assembler.

Undefined references:

E.g., file `joe.c`:

```
void foo();
int main()
{
    foo();
    return 0;
}
```

compiled on cygwin gcc

```
gcc joe.c
```

The result: an error, including something like:

... undefined symbol foo.

→	preprocessor	✓
	compiler	✓
	assembler	✓
	linker	✗

Static vs dynamic linking

Static linking

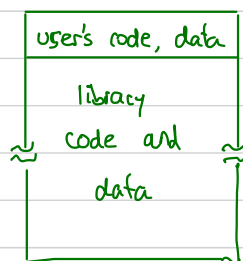
The linker copies all required library machine code and data into the executable file.

Advantages → • standalone (all code that's needed is included)

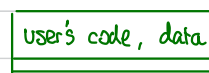
Disadvantages → • executables can be large
 - takes up disk space, memory, causes a performance hit.
 • if there are many such processes running simultaneously, could have many copies of the same library code.
 • executables must be rebuilt to use upgrades in system libraries.

Dynamic linking

Used by default on all modern OS's.



statically linked executable file.



dynamically linked executable file

info' to locate library code in the filesystem.

Advantages : → • smaller executables

- can share library machine code in memory with other programs
- executables benefit from library system upgrades.

Disadvantage: → • Some software might require a specific library version (and so fails)

Next topic: Numbers and arithmetic

Integer types: - signed and unsigned

- signed numbers: support positive and negative (and zero)
- unsigned: no negatives