Runtime Support: Loading & Linking

Loading Programs

- We've been focused on writing assembly programs. Let's turn our attention back to *running* them.
- We know that once a program is in memory, it is executed by the *fetch-execute cycle*, which is implemented in hardware.
- We know that a program called a **loader** is what copies our program into memory, but we haven't discussed this program much.
- Let's investigate how a loader works in more detail.
- We'll see one way to remove the (unrealistic) assumption that all programs must be loaded at address 0.

The Story So Far

- Up until now, we've been using a **MIPS emulator** that simulates the behaviour of a MIPS machine.
- This emulated MIPS machine has no "operating system".
- The provided program is simply loaded at address 0 and executed.
- When the program finishes, the emulator quits.
- Typically a computer would have an operating system that allows the user to select a program to run from multiple choices.
- When a program finishes, control is returned to the OS.
- Maybe the OS even allows you to run multiple programs at once!

A Simple "Operating System"

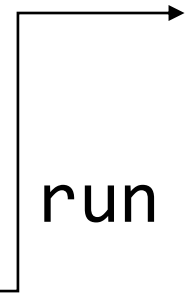
```
Operating System v1.0
repeat:
  p = next program to run
  load p into memory ──────► Loader v1.0
  jalr $0                    n = number of words in p
  beq $0, $0, repeat         for(i=0; i<n; ++i) {
                               MEM[4*i] = p[i]
                             }
```

Problems with our "OS"

- This operating system loads *all programs* at address 0.
- Where does the OS itself go in memory??
- We can't support multiple programs running at the same time.
- If we ever want to run more than one program at once, we need to be able to load programs at different addresses.
- We make the following adjustments:
  - The loader finds suitable free space in memory to load the program.
  - The loader returns the starting address of the program to the OS.
  - The loader also now sets the stack pointer in $30.

Another Simple "Operating System"

```
Operating System v2.0
repeat:
  p = next program to run
  $3 = loader(p)
  jalr $3
  beq $0, $0, repeat
```



```
Loader v2.0
loader(p) {
  n = number of words in p
  a = address of N >= 4*n
      free bytes of memory
  for(i=0; i<n; ++i) {
    MEM[a+4*i] = p[i]
  }
  $30 = a + N
  return a
}
```

Does this Work?

```
lis $20
.word 0xffff000c
lis $21
.word smileyFace
lis $4
.word 4
loop: lw $3, 0($21)
beq $3, $0, end
sw $3, 0($20)
add $21, $21, $4
beq $0, $0, loop
end: jr $31
```

```
frownyFace:
.word 0x3a ; :
.word 0x28 ; (
.word 0x0a ; newline
.word 0
smileyFace:
.word 0x3a ; :
.word 0x29 ; )
.word 0x0a ; newline
.word 0
```

This fun little program prints a smiley face :) when loaded at address 0. What could possibly go wrong when we load it at address 16 (0x10)?

Does this Work?

```
lis $20                     0x0000a014
.word 0xffff000c            0xffff000c
lis $21                     0x0000a814
.word smileyFace            0x00000040
lis $4                      0x00002014
.word 4                     0x00000004
loop: lw $3, 0($21)         0x8ea30000
beq $3, $0, end             0x10600003
sw $3, 0($20)               0xae830000
add $21, $21, $4            0x02a4a820
beq $0, $0, loop            0x1000fffb
end: jr $31                 0x03e00008
```

```
frownyFace:
.word 0x3a ; :              0x0000003a
.word 0x28 ; (              0x00000028
.word 0x0a ; newline        0x0000000a
.word 0                     0x00000000
smileyFace:
.word 0x3a ; :              0x0000003a
.word 0x29 ; )              0x00000029
.word 0x0a ; newline        0x0000000a
.word 0                     0x00000000
```

Does this Work?

```
lis $20
.word 0xffff000c
lis $21
.word smileyFace
lis $4
.word 4
loop: lw $3, 0($21)
beq $3, $0, end
sw $3, 0($20)
add $21, $21, $4
beq $0, $0, loop
end: jr $31
```

```
frownyFace:
.word 0x3a ; :
.word 0x28 ; (
.word 0x0a ; newline
.word 0
smileyFace:
.word 0x3a ; :
.word 0x29 ; )
.word 0x0a ; newline
.word 0
```

Loaded at address 0x10

Addr	Code	Addr	Code
0x10	0x0000a014	0x40	0x0000003a
0x14	0xffff000c	0x44	0x00000028
0x18	0x0000a814	0x48	0x0000000a
0x1c	0x00000040	0x4c	0x00000000
0x20	0x00002014	0x50	0x0000003a
0x24	0x00000004	0x54	0x00000029
0x28	0x8ea30000	0x58	0x0000000a
0x2c	0x10600003	0x5c	0x00000000
0x30	0xae830000		
0x34	0x02a4a820		
0x38	0x1000fffb		
0x3c	0x03e00008		

Does this Work?

```
lis $20
.word 0xffff000c
lis $21
.word smileyFace
lis $4
.word 4
loop: lw $3, 0($21)
beq $3, $0, end
sw $3, 0($20)
add $21, $21, $4
beq $0, $0, loop
end: jr $31
```

```
frownyFace:
.word 0x3a ; :
.word 0x28 ; (
.word 0x0a ; newline
.word 0
smileyFace:
.word 0x3a ; :
.word 0x29 ; )
.word 0x0a ; newline
.word 0
```

Loaded at address 0x10

Addr	Code	Addr	Code
0x10	0x0000a014	0x40	0x0000003a
0x14	0xffff000c	0x44	0x00000028
0x18	0x0000a814	0x48	0x0000000a
0x1c	0x00000040	0x4c	0x00000000
0x20	0x00002014	0x50	0x0000003a
0x24	0x00000004	0x54	0x00000029
0x28	0x8ea30000	0x58	0x0000000a
0x2c	0x10600003	0x5c	0x00000000
0x30	0xae830000		
0x34	0x02a4a820		
0x38	0x1000fffb		
0x3c	0x03e00008		

Does this Work?

```
lis $20                    frownyFace:
.word 0xffff000c           .word 0x3a ; :
lis $21                    .word 0x28 ; (
.word smileyFace           .word 0x0a ; newline
lis $4                     .word 0
.word 4                    smileyFace:
loop: lw $3, 0($21)        .word 0x3a ; :
beq $3, $0, end            .word 0x29 ; )
sw $3, 0($20)              .word 0x0a ; newline
add $21, $21, $4           .word 0
beq $0, $0, loop
end: jr $31
```

Loaded at address 0x10

Addr	Code	Addr	Code
0x10	0x0000a014	0x40	0x0000003a
0x14	0xffff000c	0x44	0x00000028
0x18	0x0000a814	0x48	0x0000000a
0x1c	0x00000040	0x4c	0x00000000
0x20	0x00002014	0x50	0x0000003a
0x24	0x00000004	0x54	0x00000029
0x28	0x8ea30000	0x58	0x0000000a
0x2c	0x10600003	0x5c	0x00000000
0x30	0xae830000		
0x34	0x02a4a820		
0x38	0x1000fffb		
0x3c	0x03e00008		

:(

Why doesn't this work?

```
lis $20
.word 0xffff000c
lis $21
.word smileyFace
lis $4
.word 4
loop: lw $3, 0($21)
beq $3, $0, end
sw $3, 0($20)
add $21, $21, $4
beq $0, $0, loop
end: jr $31
```

```
frownyFace:
.word 0x3a ; :
.word 0x28 ; (
.word 0x0a ; newline
.word 0
smileyFace:
.word 0x3a ; :
.word 0x29 ; )
.word 0x0a ; newline
.word 0
```

- Once we assemble the program, the labels in ".word label" lines are replaced with **fixed addresses.**
- But these addresses are computed relative to 0.
- If we load the program at a nonzero address, all the ".word label" lines point to the wrong place!

How Do We Fix It?

- When the loader loads the code, it also needs to perform **relocation**.
- Relocation is the process of adjusting memory addresses in the code to account for the new starting address.
- For every ".word label" line, the loader needs to offset the assembled value by the starting address it chooses for the code.
- In the previous example, the ".word smileyFace" line was assembled to 0x00000040.
- To make the program work properly when loaded at address 0x10, the loader would need to adjust this value to 0x00000050.

Why Fixing It Is Hard

- Guess what the code below assembles to:

```
jr $0
.word 8
label: .word label
```

- It assembles to:

```
0x00000008
0x00000008
0x00000008
```

- Only the third line should be relocated! Otherwise, we're messing up an instruction, or changing the value of a constant!
- But it's **impossible** to tell this by looking at the machine code.

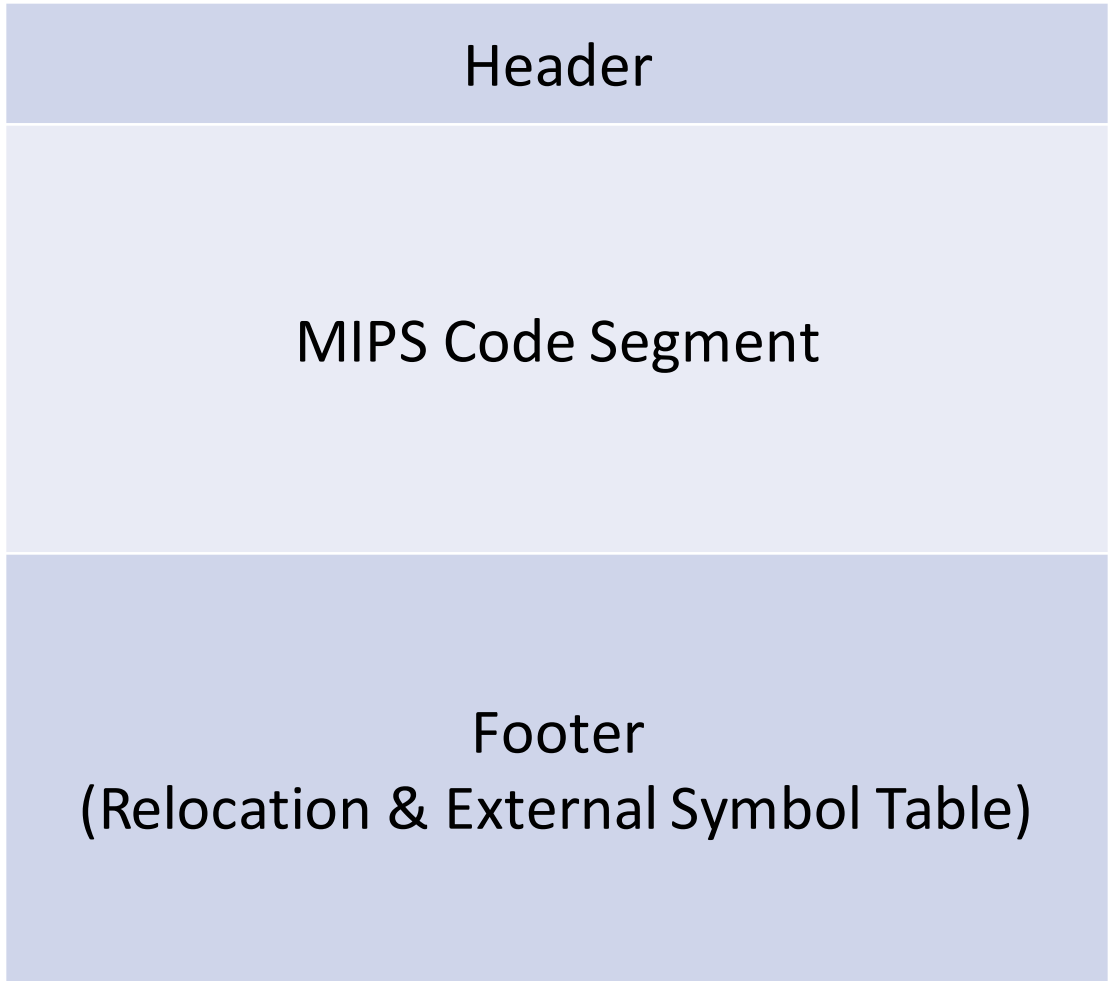
Object Code

- Relocation is impossible to do correctly unless the loader is provided with additional information.
- The loader needs to know which machine code words correspond to memory addresses (".word label" lines).
- Our solution is to modify the assembler to produce **object code** instead of raw machine code.
- An object code file combines machine code with some **metadata** about the contents of the machine code.
- The ".o" files that can be produced by g++ are object code.

The MERL File Format

- MERL (MIPS Executable Relocatable Linkable) is a simple object code format designed for this course.
- It not only contains metadata needed for relocation, but also metadata needed for **linking** (to be discussed later).
- The "executable" part of the name refers to the fact that a MERL file is a valid MIPS program and can be executed directly (at address 0).
- However, it also contains the necessary information for the loader to perform relocation and execute the code at arbitrary addresses.

MERL File Structure



- The **header** stores information about the total size of the file and the size of the MIPS code.
- The **MIPS code segment** contains the actual MIPS program (in machine code).
- The **footer** contains a table of metadata needed for relocation and linking.
- MERL is a **binary** file format. It is a sequence of 32-bit words.

The MERL Header

A MERL file's header consists of three 32-bit words:

1. The **MERL cookie**, the constant 0x10000002. This serves two purposes:
   - It is a "magic number" that helps identify the file as a valid MERL file.
   - It is the encoding of the instruction **beq $0, $0, 2**. If the MERL file is loaded into memory and executed, this instruction will skip over the rest of the header.
2. The **end of module** address, the address where the MERL file would end if loaded at address 0. (One word after the last word in the MERL file.)
   - Equivalently, this is the **total size in bytes of the MERL file**.
3. The **end of code** address, where the MIPS code segment would end (and the footer would start) if loaded at address 0.
   - Equivalently, this is the **combined size in bytes of the header and code segment**.

The MIPS Code Segment

- This portion of the file just consists of MIPS machine code, representing the program to be executed.
- An assembler that produces MERL would simply need to "wrap" the machine code it produces with the header and footer.
- There is one catch: Generating assembly code that's intended to run at address 0 **will not work** when creating MERL files.
- The header takes up 12 bytes!
- A MERL assembler needs to account for this and generate assembly code intended to run at **address 12 (0x0c)**.

The MERL Footer

- This footer can contain three types of "entries", but only one is used by the loader (we'll talk about the other two later).
- REL (Relocation) entry:
  - Specifies an address in the MIPS code corresponding to a ".word label" line.
  - REL entries specify the addresses in the MIPS code that must be relocated.
  - A REL entry consists of two 32-bit words:
    - Format code 0x00000001 (identifies this as a REL entry)
    - Address of the word that must be relocated
  - The addresses in REL entries are relative to the **MERL file** (they include the header) and assume the MERL file starts at address 0. A REL entry with address 0x0c (12) would correspond to the **first word** in the MIPS code.

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:
`lis $1`	`0x00`	`0x00000814`
`.word forward`	`0x04`	`0x00000010`
`jr $1`	`0x08`	`0x00200008`
`back: jr $31`	`0x0c`	`0x03e00008`
`forward: add $3, $0, $0`	`0x10`	`0x00001820`
`lis $1`	`0x14`	`0x00000814`
`.word back`	`0x18`	`0x0000000c`
`jr $1`	`0x1c`	`0x00200008`

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:
`lis $1`	`0x00`	`0x00000814`
`.word forward`	`0x04`	`0x00000010`
`jr $1`	`0x08`	`0x00200008`
`back: jr $31`	`0x0c`	`0x03e00008`
`forward: add $3, $0, $0`	`0x10`	`0x00001820`
`lis $1`	`0x14`	`0x00000814`
`.word back`	`0x18`	`0x0000000c`
`jr $1`	`0x1c`	`0x00200008`

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	`0x00`	`0x00000814`	`0x10000002`	`MERL cookie (beq $0, $0, 2)`
`.word forward`	`0x04`	`0x00000010`		
`jr $1`	`0x08`	`0x00200008`		
`back: jr $31`	`0x0c`	`0x03e00008`		
`forward: add $3, $0, $0`	`0x10`	`0x00001820`		
`lis $1`	`0x14`	`0x00000814`		
`.word back`	`0x18`	`0x0000000c`		
`jr $1`	`0x1c`	`0x00200008`		

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	`0x00`	`0x00000814`	`0x10000002`	`MERL cookie (beq $0, $0, 2)`
`.word forward`	`0x04`	`0x00000010`	`?? (to fill in)`	`Module end address`
`jr $1`	`0x08`	`0x00200008`		
`back: jr $31`	`0x0c`	`0x03e00008`		
`forward: add $3, $0, $0`	`0x10`	`0x00001820`		
`lis $1`	`0x14`	`0x00000814`		
`.word back`	`0x18`	`0x0000000c`		
`jr $1`	`0x1c`	`0x00200008`		

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	`0x00`	`0x00000814`	`0x10000002`	`MERL cookie (beq $0, $0, 2)`
`.word forward`	`0x04`	`0x00000010`	`?? (to fill in)`	`Module end address`
`jr $1`	`0x08`	`0x00200008`	`0x0000002c`	`Code end address`
`back: jr $31`	`0x0c`	`0x03e00008`		
`forward: add $3, $0, $0`	`0x10`	`0x00001820`		
`lis $1`	`0x14`	`0x00000814`		
`.word back`	`0x18`	`0x0000000c`		
`jr $1`	`0x1c`	`0x00200008`		

The original MIPS code ends at 0x20.
We add 0x0c for the MERL header.

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	`0x00`	`0x00000814`	`0x10000002`	`MERL cookie (beq $0, $0, 2)`
`.word forward`	`0x04`	`0x00000010`	`?? (to fill in)`	`Module end address`
`jr $1`	`0x08`	`0x00200008`	`0x0000002c`	`Code end address`
`back: jr $31`	`0x0c`	`0x03e00008`	`0x00000814`	`lis $1`
`forward: add $3, $0, $0`	`0x10`	`0x00001820`	`0x00000010`	`.word forward`
`lis $1`	`0x14`	`0x00000814`	`0x00200008`	`jr $1`
`.word back`	`0x18`	`0x0000000c`	`0x03e00008`	`back: jr $31`
`jr $1`	`0x1c`	`0x00200008`	`0x00001820`	`forward: add $3, $0, $0`
	`0x20`		`0x00000814`	`lis $1`
	`0x24`		`0x0000000c`	`.word back`
	`0x28`		`0x00200008`	`jr $1`

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	0x00	0x00000814	0x10000002	MERL cookie (beq $0, $0, 2)
`.word forward`	0x04	0x00000010	?? (to fill in)	Module end address
`jr $1`	0x08	0x00200008	0x0000002c	Code end address
`back: jr $31`	0x0c	0x03e00008	0x00000814	`lis $1`
`forward: add $3, $0, $0`	0x10	0x00001820	**0x00000010**	`.word forward`
`lis $1`	0x14	0x00000814	0x00200008	`jr $1`
`.word back`	0x18	0x0000000c	0x03e00008	`back: jr $31`
`jr $1`	0x1c	0x00200008	0x00001820	**`forward:`** `add $3, $0, $0`
	0x20		0x00000814	`lis $1`
	0x24		**0x0000000c**	`.word back`
	0x28		0x00200008	`jr $1`

These values are wrong!!
"forward" corresponds to 0x1c.
"back" corresponds to 0x18.

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	0x00	0x00000814	0x10000002	MERL cookie (beq $0, $0, 2)
`.word forward`	0x04	0x00000010	?? (to fill in)	Module end address
`jr $1`	0x08	0x00200008	0x0000002c	Code end address
`back: jr $31`	0x0c	0x03e00008	0x00000814	`lis $1`
`forward: add $3, $0, $0`	0x10	0x00001820	**0x0000001c**	`.word forward`
`lis $1`	0x14	0x00000814	0x00200008	`jr $1`
`.word back`	0x18	0x0000000c	0x03e00008	**back:** `jr $31`
`jr $1`	0x1c	0x00200008	0x00001820	**forward:** `add $3, $0, $0`
	0x20		0x00000814	`lis $1`
	0x24		**0x00000018**	`.word back`
	0x28		0x00200008	`jr $1`

Fixed.

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	0x00	0x00000814	0x10000002	MERL cookie (beq $0, $0, 2)
`.word forward`	0x04	0x00000010	?? (to fill in)	Module end address
`jr $1`	0x08	0x00200008	0x0000002c	Code end address
`back: jr $31`	0x0c	0x03e00008	0x00000814	`lis $1`
`forward: add $3, $0, $0`	0x10	0x00001820	**0x0000001c**	`.word forward`
`lis $1`	0x14	0x00000814	0x00200008	`jr $1`
`.word back`	0x18	0x0000000c	0x03e00008	**`back:`** `jr $31`
`jr $1`	0x1c	0x00200008	0x00001820	**`forward:`** `add $3, $0, $0`
	0x20		0x00000814	`lis $1`
	0x24		**0x00000018**	`.word back`
	0x28		0x00200008	`jr $1`
	0x2c		**0x00000001**	REL format code
	0x30		**0x00000010**	Address of ".word forward"

Add a relocation entry for ".word forward".

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	0x00	0x00000814	0x10000002	MERL cookie (beq $0, $0, 2)
`.word forward`	0x04	0x00000010	?? (to fill in)	Module end address
`jr $1`	0x08	0x00200008	0x0000002c	Code end address
`back: jr $31`	0x0c	0x03e00008	0x00000814	lis $1
`forward: add $3, $0, $0`	0x10	0x00001820	**0x0000001c**	.word forward
`lis $1`	0x14	0x00000814	0x00200008	jr $1
`.word back`	0x18	0x0000000c	0x03e00008	**back:** jr $31
`jr $1`	0x1c	0x00200008	0x00001820	**forward:** add $3, $0, $0
	0x20		0x00000814	lis $1
	0x24		**0x00000018**	.word back
	0x28		0x00200008	jr $1
	0x2c		**0x00000001**	REL format code
	0x30		**0x00000010**	Address of ".word forward"
	0x34		**0x00000001**	REL format code
	0x38		**0x00000024**	Address of ".word back"

Add a relocation entry for ".word back".

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	0x00	0x00000814	0x10000002	MERL cookie (beq $0, $0, 2)
`.word forward`	0x04	0x00000010	**0x0000003c**	Module end address
`jr $1`	0x08	0x00200008	0x0000002c	Code end address
`back: jr $31`	0x0c	0x03e00008	0x00000814	`lis $1`
`forward: add $3, $0, $0`	0x10	0x00001820	0x0000001c	`.word forward`
`lis $1`	0x14	0x00000814	0x00200008	`jr $1`
`.word back`	0x18	0x0000000c	0x03e00008	`back: jr $31`
`jr $1`	0x1c	0x00200008	0x00001820	`forward: add $3, $0, $0`
	0x20		0x00000814	`lis $1`
	0x24		0x00000018	`.word back`
	0x28		0x00200008	`jr $1`
	0x2c		0x00000001	REL format code
	0x30		0x00000010	Address of ".word forward"
	0x34		0x00000001	REL format code
	0x38		0x00000024	Address of ".word back"

Fill in the module end address.

MERL File Example

Assembly Source:	Address:	MIPS Machine Code:	MERL File:	Meaning:
`lis $1`	0x00	0x00000814	0x10000002	MERL cookie (beq $0, $0, 2)
`.word forward`	0x04	0x00000010	0x0000003c	Module end address
`jr $1`	0x08	0x00200008	0x0000002c	Code end address
`back: jr $31`	0x0c	0x03e00008	0x00000814	`lis $1`
`forward: add $3, $0, $0`	0x10	0x00001820	0x0000001c	`.word forward`
`lis $1`	0x14	0x00000814	0x00200008	`jr $1`
`.word back`	0x18	0x0000000c	0x03e00008	`back: jr $31`
`jr $1`	0x1c	0x00200008	0x00001820	`forward: add $3, $0, $0`
	0x20		0x00000814	`lis $1`
	0x24		0x00000018	`.word back`
	0x28		0x00200008	`jr $1`
	0x2c		0x00000001	REL format code
	0x30		0x00000010	Address of ".word forward"
	0x34		0x00000001	REL format code
	0x38		0x00000024	Address of ".word back"

MERL!

Loading with Relocation

MERL File:	Meaning:	Address:	Loaded Code:
0x10000002	MERL cookie (beq $0, $0, 2)	0x2410	
0x0000003c	Module end address	0x2414	
0x0000002c	Code end address	0x2418	
0x00000814	lis $1	0x241c	
0x0000001c	.word forward	0x2420	
0x00200008	jr $1	0x2424	
0x03e00008	back: jr $31	0x2428	
0x00001820	forward: add $3, $0, $0	0x242c	
0x00000814	lis $1		
0x00000018	.word back		
0x00200008	jr $1		
0x00000001	REL format code		
0x00000010	Address of ".word forward"		
0x00000001	REL format code		
0x00000024	Address of ".word back"		

Let's load this code at address 0x2410.

Loading with Relocation

MERL File:	Meaning:
0x10000002	**MERL cookie (beq $0, $0, 2)**
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:
0x2410	
0x2414	
0x2418	
0x241c	
0x2420	
0x2424	
0x2428	
0x242c	

The first word can be ignored (or checked to confirm this is a valid MERL file).

Loading with Relocation

MERL File:	Meaning:	Address:	Loaded Code:
0x10000002	MERL cookie (beq $0, $0, 2)	0x2410	
0x0000003c	**Module end address**	0x2414	
0x0000002c	Code end address	0x2418	
0x00000814	lis $1	0x241c	
0x0000001c	.word forward	0x2420	
0x00200008	jr $1	0x2424	
0x03e00008	back: jr $31	0x2428	
0x00001820	forward: add $3, $0, $0	0x242c	
0x00000814	lis $1		
0x00000018	.word back		
0x00200008	jr $1		
0x00000001	REL format code		
0x00000010	Address of ".word forward"		
0x00000001	REL format code		
0x00000024	Address of ".word back"		

The second word tells us how large the MERL file is. Let's keep track of it:

Variable	Value
endModule	0x3c (60)

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	**Code end address**
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:
0x2410	
0x2414	
0x2418	
0x241c	
0x2420	
0x2424	
0x2428	
0x242c	

The third word tells us how long the code segment is. Let's keep track of it as well.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	**lis $1**
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	**0x00000814**	**lis $1**
0x2414		
0x2418		
0x241c		
0x2420		
0x2424		
0x2428		
0x242c		

We read and copy the code into memory:
MEM[startAddr + index] = readWord()

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x00 (0)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	**.word forward**
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	**0x0000001c**	**.word forward**
0x2418		
0x241c		
0x2420		
0x2424		
0x2428		
0x242c		

We're not relocating ".word label" lines (yet).

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x04 (4)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	**jr $1**
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	**0x00200008**	**jr $1**
0x241c		
0x2420		
0x2424		
0x2428		
0x242c		

We read and copy the code into memory:
MEM[startAddr + index] = readWord()

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x08 (8)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	**back: jr $31**
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	**0x03e00008**	**back: jr $31**
0x2420		
0x2424		
0x2428		
0x242c		

We read and copy the code into memory:
MEM[startAddr + index] = readWord()

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x0c (12)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	**forward: add $3, $0, $0**
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	**0x00001820**	**forward: add $3, $0, $0**
0x2424		
0x2428		
0x242c		

We read and copy the code into memory:
MEM[startAddr + index] = readWord()

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x10 (16)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	**lis $1**
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	**0x00000814**	**lis $1**
0x2428		
0x242c		

We read and copy the code into memory:
MEM[startAddr + index] = readWord()

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x14 (20)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	**.word back**
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	**0x00000018**	**.word back**
0x242c		

We read and copy the code into memory:
MEM[startAddr + index] = readWord()

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x18 (24)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	**jr $1**
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	**0x00200008**	**jr $1**

We read and copy the code into memory:
MEM[startAddr + index] = readWord()

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x1c (28)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

The code is in memory. Now we read the REL entries and modify the code.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	0x20 (32)

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

We reuse the "index" variable to determine when we're done reading the table (index == endModule).

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	**REL format code**
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

REL entry detected.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	**Address of ".word forward"**
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

The address to relocate, relative to the code segment in the MERL file, is 0x10.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	**Address of ".word forward"**
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

So in the raw MIPS code (without MERL header) the address to relocate was 0x04. (0x10 – 0x0c)

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	**Address of ".word forward"**
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x0000001c	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

So in the loaded code we should relocate the word at 0x2414. (0x10 – 0x0c + 0x2410)

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	**Address of ".word forward"**
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	**0x0000001c**	**.word forward**
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

So do we change 0x1c to 0x242c (add 0x2410)? No, that doesn't work.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	**Address of ".word forward"**
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	**0x0000001c**	**.word forward**
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

The header was removed from the loaded code! So we do 0x1c + 0x2410 – 0x0c to get 0x2420 (correct).

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	**Address of ".word forward"**
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	**0x00002420**	**.word forward**
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

MEM[startAddr + relAddr – 0x0c] += startAddr – 0x0c

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x2c (44)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

Increment index by **8** (because we processed two words) and move on to the next entry.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x34 (52)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	**REL format code**
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

Another REL entry.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x34 (52)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	**Address of ".word back"**

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

MEM[startAddr + relAddr – 0x0c] += startAddr – 0x0c

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x34 (52)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	**Address of ".word back"**

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

MEM[0x2410 + 0x24 – 0x0c] += 0x2410 – 0x0c

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x34 (52)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	**Address of ".word back"**

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x00000018	.word back
0x242c	0x00200008	jr $1

MEM[0x2428] += 0x2404

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x34 (52)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	**Address of ".word back"**

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	**0x0000241c**	**.word back**
0x242c	0x00200008	jr $1

MEM[0x2428] += 0x2404

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x34 (52)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x0000241c	.word back
0x242c	0x00200008	jr $1

Increment index by **8**.

Variable	Value
endModule	0x3c (60)
endCode	0x2c (44)
index	**0x3c (60)**

Loading with Relocation

MERL File:	Meaning:
0x10000002	MERL cookie (beq $0, $0, 2)
0x0000003c	Module end address
0x0000002c	Code end address
0x00000814	lis $1
0x0000001c	.word forward
0x00200008	jr $1
0x03e00008	back: jr $31
0x00001820	forward: add $3, $0, $0
0x00000814	lis $1
0x00000018	.word back
0x00200008	jr $1
0x00000001	REL format code
0x00000010	Address of ".word forward"
0x00000001	REL format code
0x00000024	Address of ".word back"

Address:	Loaded Code:	
0x2410	0x00000814	lis $1
0x2414	0x00002420	.word forward
0x2418	0x00200008	jr $1
0x241c	0x03e00008	back: jr $31
0x2420	0x00001820	forward: add $3, $0, $0
0x2424	0x00000814	lis $1
0x2428	0x0000241c	.word back
0x242c	0x00200008	jr $1

Since index == endModule, we have processed the whole MERL file and we're done.

Variable	Value
endModule	**0x3c (60)**
endCode	0x2c (44)
index	**0x3c (60)**

Loading with Relocation: Pseudocode

```
Loader v3.0
cookie = readWord()
endModule = readWord()
endCode = readWord()
codeSize = endCode – 0x0c
startAddr = address of N >= [codeSize]
            free bytes of memory
for(index=0; index<codeSize; index+=4) {
  MEM[startAddr + index] = readWord()
}
// code is loaded into memory
// time to relocate
index = endCode
while(index<endModule) {
  formatCode = readWord()
  if(formatCode == 1) { // REL entry
     relAddr = readWord()
     MEM[startAddr + relAddr – 0x0c]
        += startAddr – 0x0c
     index += 8
  }
}
$30 = startAddr + N
return startAddr
```

Loading: Summary

- Loading all programs at address 0 is not really practical.
- But loading programs at nonzero addresses breaks label references of the form ".word label". They need to be **relocated**.
- This is impossible to do correctly just by looking at raw machine code.
- We upgrade our assembler to produce **object code** with metadata needed for relocation (and later, linking).
- Our object code format is called MERL. The loader can use the metadata in MERL files (REL entries) to properly relocate code.
- You should never **hardcode** memory addresses because REL entries won't be generated for them in a MERL file! Use labels.

Linking

- We'll now look at a rather different problem, but one that's also solved using our MERL object code format.
- Most high-level languages have a *standard library* of useful procedures that any program can use. What's the best way to implement this?
- We need a way to use the library procedures in our programs.
- The simplest way is to just put all the required code in a single file.
  - You could copy and paste everything in manually.
  - You could use **cat** or a similar tool to concatenate the programs.
  - You could add a feature similar to C/C++ #include directives to the assembler.

Problems With This "Simple" Method

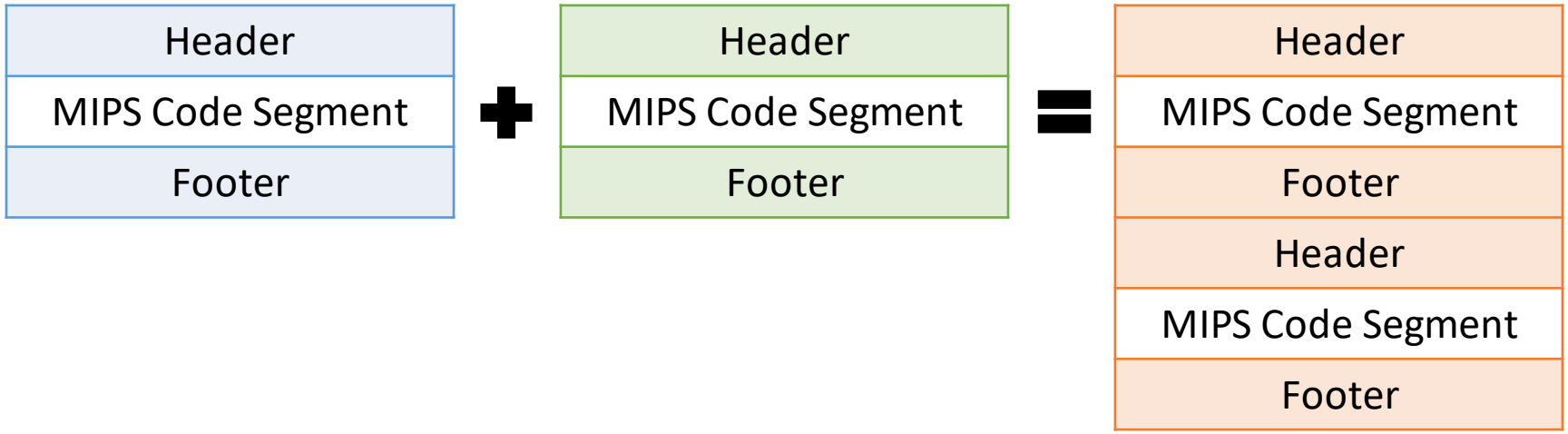
1. All assembly code has to be made available as source code.
   - The creators of libraries often do not want to distribute the source code, especially if it is provided as a commercial product.
2. All assembly code has to be assembled at once.
   - Imagine a large real-world program where the process of assembly takes a significant amount of time.
   - The libraries are probably not going to change very often, so it would be useful if we could use pre-assembled versions.
3. Duplicate label definitions become a huge pain.
   - Imagine the library writer used the label name "loop" somewhere. Now you are not allowed to use this label in your program.

Solution

- We could try combining MIPS machine code files instead of assembly source code…
- But MIPS machine code has no label information.
- If one file refers to a label defined in another file, the reference cannot be resolved.
- We need to combine the machine code with **metadata** about the labels that were defined and referenced in the original source code.
- We need to combine **MERL files**!

Combining MERL Files

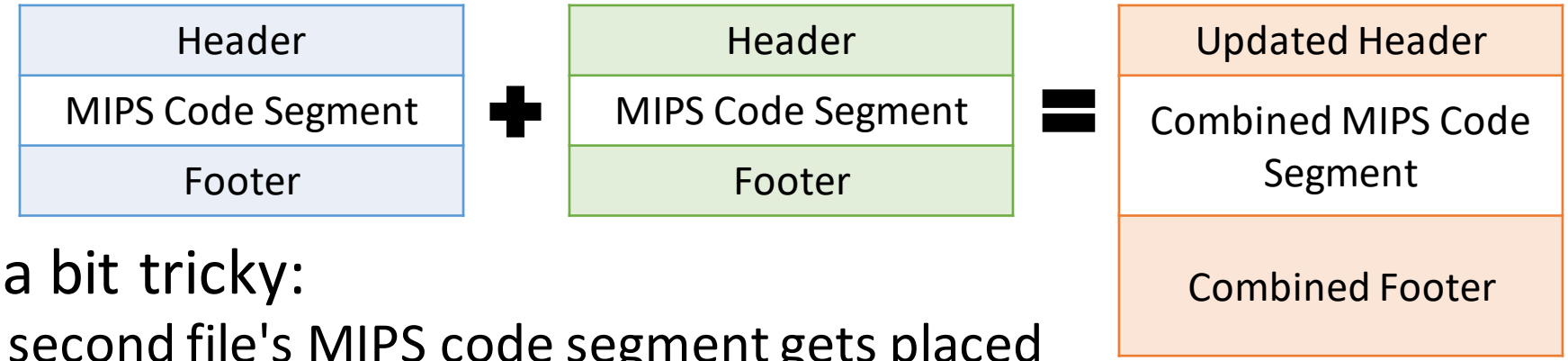
- But we can't just blindly concatenate MERL files.



- This isn't a valid MERL file!
- We need a "smart concatenation" algorithm: **linking**.

Linking

- Linking produces a valid MERL file for the combined program.



- This is a bit tricky:
  - The second file's MIPS code segment gets placed after the first one. It starts at a new location, so it must be **relocated**!
  - The footers cannot just be merged. The second footer's entries must be updated to refer to the **relocated** second code segment.
  - The header must be updated too.
  - Finally, the linker must **resolve cross-file label references!**

Import and Export Directives

- To facilitate linking, we introduce two new elements to our MIPS assembly language.
- The **.export** directive lets us export a label, making it visible to other files even after assembly (adding it to the object code metadata).
- The **.import** directive lets us import a label from an external object code file, provided it has been exported.
- The .import directive creates metadata entries for each *use* of the external label, indicating that it is waiting for the right address to be filled in at a certain location. The linker fills in the address using the .export information.

Structure of ESR and ESD Entries

- ESR entry: 3+n words
  - Word 1: 0x00000011 *[ESR format code]*
  - Word 2: *[Address of label reference]*
  - Word 3: *[Length n of the label name]*
  - Words 4–end: *[One word for each ASCII character of the label name]*

- ESD entry: 3+n words
  - Word 1: 0x00000005 *[ESD format code]*
  - Word 2: *[Address of label definition]*
  - Word 3: *[Length n of the label name]*
  - Words 4–end: *[One word for each ASCII character of the label name]*

External Symbol Reference

Generated when a label is referred to with a ".word label" directive, and the label is imported from another file with a ".import label" directive.

Stores the address of the ".word label" line in the MERL file, and the name of the label.

External Symbol Definition

Generated when a label is defined using the "label:" syntax, and the label is also exported with a ".export label" directive.

Stores the address in the MERL file corresponding the "label:" definition, and the name of the label.

Example

- Suppose we have these two separate programs:

```
.import kitten
lis $13
.word kitten
jr $13
```

```
.export kitten
kitten:
add $13, $13, $13
jr $31
```

- If these programs are converted into MERL and linked, the resulting MERL file should be equivalent to a MERL file for the following program:

```
.export kitten
lis $13
.word kitten
jr $13
kitten:
add $13, $13, $13
jr $31
```

Resolving External Label References

- Using the .import and .export directives, MIPS assembly source files can refer to labels from other files!
- To understand what the linker needs to do to resolve these references, let's take a closer look at how the assembler handles this.
- Normally, using a label that is not defined in the same file is an error.
- For a MERL assembler, there's a special exception to this rule: if the label is not defined, *but a .import directive exists for it*, this is valid.
  - Exception to the exception: You can only use imported labels in ".word label" directives, not in branch instructions.
- The assembler has no idea what address it should use for the label, so it just uses 0 as a placeholder value.

Resolving External Label References

- For each ".word label" directive where the label is imported, a zero word is generated as a placeholder.
- The linker has the necessary information to **fill in** this placeholder word with the correct value.
- The ESR (External Symbol Reference) and ESD (External Symbol Definition) entries in the MERL file allow the linker to calculate which locations should be modified, and what the new value should be.
- Getting the calculations correct is somewhat complicated, since the linker also relocates the second code segment.

Resolving External Label References

Algorithm to resolve external label references for files "M1" and "M2":

1. Loop over the label imports in M1's footer.
2. For each import, check if M2's footer contains a matching export (the imported label and exported label have the same name).
3. If so, replace the word of M1's code segment corresponding to the imported label with the address of the exported label.
4. *Change the ESR entry to a REL entry*, because the address of the import no longer contains a placeholder, it now contains a relocatable value!
5. Repeat Steps 1 to 4 with the roles of M1 and M2 switched.

The necessary information (label names, addresses of imports, addresses of exports) is in the MERL footer's ESR and ESD entries.

Why Do ESRs Become RELs?

```
.import kitten
lis $13
.word kitten
jr $13
```

```
.export kitten
kitten:
add $13, $13, $13
jr $31
```

```
.export kitten
lis $13
.word kitten
jr $13
kitten:
add $13, $13, $13
jr $31
```

```
ESR entry
for ".word kitten"
```

```
ESD entry
for "kitten:"
```

```
REL entry
for ".word kitten"
ESD entry
for "kitten:"
```

If we don't change the ESR to a REL, then ".word kitten" won't be relocated properly.

Linking Algorithm: Overview

1. Check for duplicate exports (M1 and M2 export the same label).
2. Combine M1's code segment and M2's code segment.

Relocate M2's code segment. This involves two steps:

3. Determine the shift in starting address that resulted when M2's code was moved after M1's code. Update all the addresses in M2's footer (all addresses contained in REL, ESR and ESD entries) according to this shift.
4. Use the updated REL entries in M2's footer to actually relocate M2's code.

5. Resolve external symbol references (imports) for M1.
6. Resolve external symbol references (imports) for M2.
7. Combine M1's footer with M2's footer.
8. Construct the linked header using the sizes of the linked code segment and linked footer.
9. Output the complete linked MERL file to standard output.

Linking Algorithm: Example

M1.asm		M2.asm	
.import foo		.export foo	
.export bar		.import bar	
sw $31, -4($30)		foo: lis $2	
lis $29		.word -1	
.word foo		lis $28	
jalr $29		.word loop	
lis $3		lis $29	
.word bar		.word bar	
lw $3, 0($3)		loop: lw $3, 0($29)	
lw $31, -4($30)		add $3, $1, $3	
jr $31		sw $3, 0($29)	
bar: .word 0		add $1, $1, $2	
		bne $1, $0, skip	
		jr $31	
		skip: jr $28	

Linking Algorithm: Example

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000018 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000020 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x0000000c (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Step 1: Check for duplicate label export errors.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000018 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000020 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x0000000c (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

There are none. (ESDs have different names)

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000018 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000020 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x0000000c (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Done Step 1.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000018 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000020 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x0000000c (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Step 2: Combine the code segments.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000018 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000020 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x0000000c (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Step 2: Combine the code segments.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x00000024 (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

Done Step 2.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x00000024 (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

Steps 3 & 4: Relocate the code segment from M2.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x00000024 (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

M2's code segment originally started at 0x0c (12).

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000018 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000020 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x0000000c (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Now it starts at 0x34 (52). Difference of 40 (0x28).

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x00000024 (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

Step 3: Offset the addresses in M2's table by 0x28.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000018 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000020 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x0000000c (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Step 3: Offset the addresses in M2's table by 0x28.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Done Step 3.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Step 4: Relocate M2's code segment in LINKED using the new table.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x00000024 (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

M2.merl			
0x00	0x10000002	0x3c	0x03800008
0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x03e00008	0x74	0x0000006f (o)

Step 4: Relocate M2's code segment in LINKED using the new table.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x00000024 (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

M2.merl			
0x00	0x10000002	0x3c	0x03800008
0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x03e00008	0x74	0x0000006f (o)

Add 0x28 to the relocatable value in the linked code.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x0000004c (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

M2.merl			
0x00	0x10000002	0x3c	0x03800008
0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x03e00008	0x74	0x0000006f (o)

Done Step 4.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x0000004c (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000000 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

M2.merl			
0x00	0x10000002	0x3c	0x03800008
0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x03e00008	0x74	0x0000006f (o)

Step 5: Resolve imports in M1 using exports from M2.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Label "foo" is imported by M1.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

M2 has a matching export of "foo".

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Set the word at address 0x14 in LINKED to the value 0x34.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000011 (ESR)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000003 (nameLen)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000066 (f)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x0000006f (o)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x0000006f (o)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000005 (ESD)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000030 (address)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)	0x5c	0x00000003 (nameLen)	0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000	0x60	0x00000062 (b)	0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc	0x64	0x00000061 (a)	0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008	0x68	0x00000072 (r)	0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Set the word at address 0x14 in LINKED to the value 0x34.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x0000004c (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000000 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000034 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

The ESR in M1 is resolved and becomes a REL.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Done Step 5.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Step 6: Resolve imports in M2 using exports from M1.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

M2 imports "bar".

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

M1 has a matching export of "bar".

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Set the word at address 0x48 in LINKED to the value 0x30.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000011 (ESR)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000062 (b)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000061 (a)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000072 (r)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000005 (ESD)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x00000034 (address)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x00000003 (nameLen)
0x30	0x00000000 (bar:)			0x30	0x00220820	0x6c	0x00000066 (f)
0x34	0x00000001 (REL)			0x34	0x14200001	0x70	0x0000006f (o)
0x38	0x00000020 (address)			0x38	0x03e00008	0x74	0x0000006f (o)

Set the word at address 0x48 in LINKED to the value 0x30.

LINKED.merl			
0x00	0x10000002	0x3c	0x0000e014
0x04	??? (endModule)	0x40	0x0000004c (.word loop)
0x08	0x00000068 (endCode)	0x44	0x0000e814
0x0c	0xafdffffc	0x48	0x00000030 (.word bar)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)
0x14	0x00000034 (.word foo)	0x50	0x00231820
0x18	0x03a00009	0x54	0xafa30000
0x1c	0x00001814	0x58	0x00220820
0x20	0x00000030 (.word bar)	0x5c	0x14200001
0x24	0x8c630000	0x60	0x03e00008
0x28	0x8fdffffc	0x64	0x03800008
0x2c	0x03e00008		
0x30	0x00000000 (bar:)		
0x34	0x00001014 (foo:)		
0x38	0xffffffff		

The ESR in M2 is resolved and becomes a REL.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000001 (REL)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000005 (ESD)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000034 (address)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000003 (nameLen)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000066 (f)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x0000006f (o)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x0000006f (o)
0x30	0x00000000 (bar:)			0x30	0x00220820		
0x34	0x00000001 (REL)			0x34	0x14200001		
0x38	0x00000020 (address)			0x38	0x03e00008		

Done Step 6.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000001 (REL)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000005 (ESD)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000034 (address)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000003 (nameLen)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000066 (f)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x0000006f (o)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x0000006f (o)
0x30	0x00000000 (bar:)			0x30	0x00220820		
0x34	0x00000001 (REL)			0x34	0x14200001		
0x38	0x00000020 (address)			0x38	0x03e00008		

Step 7: Combine the footers into a linked footer.

M1.merl				M2.merl			
0x00	0x10000002	0x3c	0x00000001 (REL)	0x00	0x10000002	0x3c	0x03800008
0x04	0x0000006c (endModule)	0x40	0x00000014 (address)	0x04	0x00000078 (endModule)	0x40	0x00000001 (REL)
0x08	0x00000034 (endCode)	0x44	0x00000005 (ESD)	0x08	0x00000040 (endCode)	0x44	0x00000040 (address)
0x0c	0xafdffffc	0x48	0x00000030 (address)	0x0c	0x00001014 (foo:)	0x48	0x00000001 (REL)
0x10	0x0000e814	0x4c	0x00000003 (nameLen)	0x10	0xffffffff	0x4c	0x00000048 (address)
0x14	0x00000000 (.word foo)	0x50	0x00000062 (b)	0x14	0x0000e014	0x50	0x00000003 (nameLen)
0x18	0x03a00009	0x54	0x00000061 (a)	0x18	0x00000024 (.word loop)	0x54	0x00000005 (ESD)
0x1c	0x00001814	0x58	0x00000072 (r)	0x1c	0x0000e814	0x58	0x00000034 (address)
0x20	0x00000030 (.word bar)			0x20	0x00000000 (.word bar)	0x5c	0x00000003 (nameLen)
0x24	0x8c630000			0x24	0x8fa30000 (loop:)	0x60	0x00000066 (f)
0x28	0x8fdffffc			0x28	0x00231820	0x64	0x0000006f (o)
0x2c	0x03e00008			0x2c	0xafa30000	0x68	0x0000006f (o)
0x30	0x00000000 (bar:)			0x30	0x00220820		
0x34	0x00000001 (REL)			0x34	0x14200001		
0x38	0x00000020 (address)			0x38	0x03e00008		

Step 7: Combine the footers into a linked footer.

LINKED.merl							
0x00	0x10000002	0x3c	0x0000e014	0x78	0x00000005 (ESD)	0xa0	0x00000005 (ESD)
0x04	??? (endModule)	0x40	0x0000004c (.word loop)	0x7c	0x00000030 (address)	0xa4	0x00000034 (address)
0x08	0x00000068 (endCode)	0x44	0x0000e814	0x80	0x00000003 (nameLen)	0xa8	0x00000003 (nameLen)
0x0c	0xafdffffc	0x48	0x00000030 (.word bar)	0x84	0x00000062 (b)	0xac	0x00000066 (f)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)	0x88	0x00000061 (a)	0xb0	0x0000006f (o)
0x14	0x00000034 (.word foo)	0x50	0x00231820	0x8c	0x00000072 (r)	0xb4	0x0000006f (o)
0x18	0x03a00009	0x54	0xafa30000	0x90	0x00000001 (REL)		
0x1c	0x00001814	0x58	0x00220820	0x94	0x00000040 (address)		
0x20	0x00000030 (.word bar)	0x5c	0x14200001	0x98	0x00000001 (REL)		
0x24	0x8c630000	0x60	0x03e00008	0x9c	0x00000048 (address)		
0x28	0x8fdffffc	0x64	0x03800008				
0x2c	0x03e00008	0x68	0x00000001 (REL)				
0x30	0x00000000 (bar:)	0x6c	0x00000020 (address)				
0x34	0x00001014 (foo:)	0x70	0x00000001 (REL)				
0x38	0xffffffff	0x74	0x00000014 (address)				

Done Step 7.

LINKED.merl							
0x00	0x10000002	0x3c	0x0000e014	0x78	0x00000005 (ESD)	0xa0	0x00000005 (ESD)
0x04	??? (endModule)	0x40	0x0000004c (.word loop)	0x7c	0x00000030 (address)	0xa4	0x00000034 (address)
0x08	0x00000068 (endCode)	0x44	0x0000e814	0x80	0x00000003 (nameLen)	0xa8	0x00000003 (nameLen)
0x0c	0xafdffffc	0x48	0x00000030 (.word bar)	0x84	0x00000062 (b)	0xac	0x00000066 (f)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)	0x88	0x00000061 (a)	0xb0	0x0000006f (o)
0x14	0x00000034 (.word foo)	0x50	0x00231820	0x8c	0x00000072 (r)	0xb4	0x0000006f (o)
0x18	0x03a00009	0x54	0xafa30000	0x90	0x00000001 (REL)		
0x1c	0x00001814	0x58	0x00220820	0x94	0x00000040 (address)		
0x20	0x00000030 (.word bar)	0x5c	0x14200001	0x98	0x00000001 (REL)		
0x24	0x8c630000	0x60	0x03e00008	0x9c	0x00000048 (address)		
0x28	0x8fdffffc	0x64	0x03800008				
0x2c	0x03e00008	0x68	0x00000001 (REL)				
0x30	0x00000000 (bar:)	0x6c	0x00000020 (address)				
0x34	0x00001014 (foo:)	0x70	0x00000001 (REL)				
0x38	0xffffffff	0x74	0x00000014 (address)				

Step 8: Fill in the header.

LINKED.merl							
0x00	0x10000002	0x3c	0x0000e014	0x78	0x00000005 (ESD)	0xa0	0x00000005 (ESD)
0x04	??? (endModule)	0x40	0x0000004c (.word loop)	0x7c	0x00000030 (address)	0xa4	0x00000034 (address)
0x08	0x00000068 (endCode)	0x44	0x0000e814	0x80	0x00000003 (nameLen)	0xa8	0x00000003 (nameLen)
0x0c	0xafdffffc	0x48	0x00000030 (.word bar)	0x84	0x00000062 (b)	0xac	0x00000066 (f)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)	0x88	0x00000061 (a)	0xb0	0x0000006f (o)
0x14	0x00000034 (.word foo)	0x50	0x00231820	0x8c	0x00000072 (r)	0xb4	0x0000006f (o)
0x18	0x03a00009	0x54	0xafa30000	0x90	0x00000001 (REL)		
0x1c	0x00001814	0x58	0x00220820	0x94	0x00000040 (address)		
0x20	0x00000030 (.word bar)	0x5c	0x14200001	0x98	0x00000001 (REL)		
0x24	0x8c630000	0x60	0x03e00008	0x9c	0x00000048 (address)		
0x28	0x8fdffffc	0x64	0x03800008				
0x2c	0x03e00008	0x68	0x00000001 (REL)				
0x30	0x00000000 (bar:)	0x6c	0x00000020 (address)				
0x34	0x00001014 (foo:)	0x70	0x00000001 (REL)				
0x38	0xffffffff	0x74	0x00000014 (address)				

Step 8: Fill in the header.

LINKED.merl							
0x00	0x10000002	0x3c	0x0000e014	0x78	0x00000005 (ESD)	0xa0	0x00000005 (ESD)
0x04	0x000000b8 (endModule)	0x40	0x0000004c (.word loop)	0x7c	0x00000030 (address)	0xa4	0x00000034 (address)
0x08	0x00000068 (endCode)	0x44	0x0000e814	0x80	0x00000003 (nameLen)	0xa8	0x00000003 (nameLen)
0x0c	0xafdffffc	0x48	0x00000030 (.word bar)	0x84	0x00000062 (b)	0xac	0x00000066 (f)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)	0x88	0x00000061 (a)	0xb0	0x0000006f (o)
0x14	0x00000034 (.word foo)	0x50	0x00231820	0x8c	0x00000072 (r)	0xb4	0x0000006f (o)
0x18	0x03a00009	0x54	0xafa30000	0x90	0x00000001 (REL)		
0x1c	0x00001814	0x58	0x00220820	0x94	0x00000040 (address)		
0x20	0x00000030 (.word bar)	0x5c	0x14200001	0x98	0x00000001 (REL)		
0x24	0x8c630000	0x60	0x03e00008	0x9c	0x00000048 (address)		
0x28	0x8fdffffc	0x64	0x03800008				
0x2c	0x03e00008	0x68	0x00000001 (REL)				
0x30	0x00000000 (bar:)	0x6c	0x00000020 (address)				
0x34	0x00001014 (foo:)	0x70	0x00000001 (REL)				
0x38	0xffffffff	0x74	0x00000014 (address)				

Done Step 8.

LINKED.merl							
0x00	0x10000002	0x3c	0x0000e014	0x78	0x00000005 (ESD)	0xa0	0x00000005 (ESD)
0x04	0x000000b8 (endModule)	0x40	0x0000004c (.word loop)	0x7c	0x00000030 (address)	0xa4	0x00000034 (address)
0x08	0x00000068 (endCode)	0x44	0x0000e814	0x80	0x00000003 (nameLen)	0xa8	0x00000003 (nameLen)
0x0c	0xafdffffc	0x48	0x00000030 (.word bar)	0x84	0x00000062 (b)	0xac	0x00000066 (f)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)	0x88	0x00000061 (a)	0xb0	0x0000006f (o)
0x14	0x00000034 (.word foo)	0x50	0x00231820	0x8c	0x00000072 (r)	0xb4	0x0000006f (o)
0x18	0x03a00009	0x54	0xafa30000	0x90	0x00000001 (REL)		
0x1c	0x00001814	0x58	0x00220820	0x94	0x00000040 (address)		
0x20	0x00000030 (.word bar)	0x5c	0x14200001	0x98	0x00000001 (REL)		
0x24	0x8c630000	0x60	0x03e00008	0x9c	0x00000048 (address)		
0x28	0x8fdffffc	0x64	0x03800008				
0x2c	0x03e00008	0x68	0x00000001 (REL)				
0x30	0x00000000 (bar:)	0x6c	0x00000020 (address)				
0x34	0x00001014 (foo:)	0x70	0x00000001 (REL)				
0x38	0xffffffff	0x74	0x00000014 (address)				

Step 9: Output the linked MERL file, and we're done with linking!

LINKED.merl							
0x00	0x10000002	0x3c	0x0000e014	0x78	0x00000005 (ESD)	0xa0	0x00000005 (ESD)
0x04	0x000000b8 (endModule)	0x40	0x0000004c (.word loop)	0x7c	0x00000030 (address)	0xa4	0x00000034 (address)
0x08	0x00000068 (endCode)	0x44	0x0000e814	0x80	0x00000003 (nameLen)	0xa8	0x00000003 (nameLen)
0x0c	0xafdffffc	0x48	0x00000030 (.word bar)	0x84	0x00000062 (b)	0xac	0x00000066 (f)
0x10	0x0000e814	0x4c	0x8fa30000 (loop:)	0x88	0x00000061 (a)	0xb0	0x0000006f (o)
0x14	0x00000034 (.word foo)	0x50	0x00231820	0x8c	0x00000072 (r)	0xb4	0x0000006f (o)
0x18	0x03a00009	0x54	0xafa30000	0x90	0x00000001 (REL)		
0x1c	0x00001814	0x58	0x00220820	0x94	0x00000040 (address)		
0x20	0x00000030 (.word bar)	0x5c	0x14200001	0x98	0x00000001 (REL)		
0x24	0x8c630000	0x60	0x03e00008	0x9c	0x00000048 (address)		
0x28	0x8fdffffc	0x64	0x03800008				
0x2c	0x03e00008	0x68	0x00000001 (REL)				
0x30	0x00000000 (bar:)	0x6c	0x00000020 (address)				
0x34	0x00001014 (foo:)	0x70	0x00000001 (REL)				
0x38	0xffffffff	0x74	0x00000014 (address)				