Symbol Table

<table>
<thead>
<tr>
<th>Label</th>
<th>Section</th>
<th>Byte Offset</th>
<th>Local/Global</th>
<th>Type</th>
<th>Label Sequence #</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcGreeting</td>
<td>rodata</td>
<td>0</td>
<td>local</td>
<td>?</td>
<td>0</td>
</tr>
<tr>
<td>main</td>
<td>text</td>
<td>0</td>
<td>global</td>
<td>function</td>
<td>1</td>
</tr>
<tr>
<td>printf</td>
<td>?</td>
<td>7</td>
<td>local</td>
<td>?</td>
<td>2</td>
</tr>
</tbody>
</table>

Relocation Records

<table>
<thead>
<tr>
<th>Section</th>
<th>Byte Offset</th>
<th>Relocation Type</th>
<th>Label Sequence #</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>3</td>
<td>32 bit absolute</td>
<td>0</td>
</tr>
<tr>
<td>text</td>
<td>8</td>
<td>32 bit displacement</td>
<td>2</td>
</tr>
</tbody>
</table>

Rodata Section

<table>
<thead>
<tr>
<th>Byte Offset</th>
<th>Contents (hex)</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>48</td>
<td>.asciz &quot;Hello\n&quot;</td>
</tr>
<tr>
<td>1</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6C</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6C</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6F</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0A</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>00</td>
<td></td>
</tr>
</tbody>
</table>
### Byte Offset | Contents (hex) | Explanation
---|---|---
0 | 55 | `pushl %ebp` <br> 01010101 <br>This is a “pushl %ebp” instruction
1-2 | 89 E5 | `movl %esp, %ebp` 10001001 11 100 101 <br>This is a “movl” instruction whose source operand is a register <br>The M field designates a register <br>The source register is %esp <br>The destination register is %ebp
3-7 | 68??????? | `pushl $pcGreeting` 01101000 ??????????????????????????????? <br>This is a “pushl” instruction with a 4 byte immediate operand <br>This is the data to be pushed
8-12 | E8??????? | `call printf` 11101000 ??????????????????????????????? <br>This is a “call” instruction with a 4 byte immediate operand <br>This is the displacement to the instruction to be called.
13-15 | 83C404 | `addl $4, %esp` 10000011 11 000 100 00000100 <br>This is some “l” instruction that has a 1 byte immediate operand <br>The M field designates a register <br>This is an “add” instruction <br>The destination register is %esp <br>The immediate operand is 4
16-20 | B800000000 | `movl $0, %eax` 10111000 00000000000000000000000000000000 <br>This is an instruction of the form “movl 4-byte-immediate, %eax” <br>The immediate operand is 0
21-22 | 89EC | `movl %ebp, %esp` 10001001 11 101 100 <br>This is a “movl” instruction whose source operand is a register <br>The M field designates a register <br>The source register is %ebp <br>The destination register is %esp
23 | 5D | `popl %ebp` 01011101 <br>This is a “popl %ebp” instruction
24 | C3 | `ret` 11000011 <br>This is a “ret” instruction