Objtool on Power - Update
Livepatching Miniconf - Linux Plumbers 2019

Kamalesh Babulal
Linux Technology Center, IBM
kamalesh@linux.ibm.com
Objtool on Power - Update

Current State

https://github.com/kamalesh-babulal/linux/tree/objtool-v1

- Decoder
  - Stack operations
  - Branch
  - Dead ends
  - Switch table (solution - gcc plugin)
  - Conditional branch to LR
  - Prologue Sequence
  - Validation of stack

Kernel
- Annotation – reachable/unreachable
- ASM Cleanup

Future Plans

- Once ARM64 objtool code is merged
- It decouples arch independent bits
- Patchset is based on these changes
- We can do more clean up
  - Structures (registers)
- Common Plugin for switch table detection
- More and more testing required
- Try different versions of gcc
- Tricky instructions with ASM
This work represents the view of the authors and does not necessarily represent the view of their employers.

IBM and IBM (logo) are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product, and service names may be trademarks or service marks of others.
Thank you
Backup
Stack Frame on PPC64

Figure 2-18. Stack Frame Organization
Switch case representation

Figure 2-37. **Absolute Switch Code (Within)** for static modules located in low or high 2 GB of address space

<table>
<thead>
<tr>
<th>C Code</th>
<th>Assembly Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch(j)</td>
<td>cmplwi r12, 4</td>
</tr>
<tr>
<td></td>
<td>bge .Ldefault</td>
</tr>
<tr>
<td>case 0:</td>
<td>slwi r12, 2</td>
</tr>
<tr>
<td></td>
<td>addis r12, r12, .Ltab@ha</td>
</tr>
<tr>
<td>case 1:</td>
<td>lwa r12, .Ltab@l(r12)</td>
</tr>
<tr>
<td></td>
<td>mtctr r12</td>
</tr>
<tr>
<td>case 3:</td>
<td>bcctr</td>
</tr>
<tr>
<td>default:</td>
<td>.rodata</td>
</tr>
<tr>
<td></td>
<td>.Ltab:</td>
</tr>
<tr>
<td></td>
<td>... .long .Lcase0</td>
</tr>
<tr>
<td>}</td>
<td>.long .Lcase1</td>
</tr>
<tr>
<td></td>
<td>.long .Ldefault</td>
</tr>
<tr>
<td></td>
<td>.long .Lcase3</td>
</tr>
<tr>
<td></td>
<td>.text</td>
</tr>
</tbody>
</table>
Prologue Sequence

0000000000000000 <rdma_event_msg-0x8>
...
0: R_PPC64_REL64 .TOC.-0x8
0000000000000008 <rdma_event_msg>:
  8: f8 ff 4c e8  ld  r2,-8(r12)
  8: R_PPC64_ENTRY  *ABS*