GP relaxation in LLD

Using ‘lwgp‘ as example

Wu Xinlong

PLCT Lab

Dec. 15, 2021

```
1 lui  a0, %hi(Ig)
2 lw   a1, %lo(Ig)(a0)
```

```
1 lui  a0, %hi(Ig)
2 lw   a1, %lo(Ig)(a0)
```

```
1 lui  a0, 0x20000
2 lw   a1, 1024(a0)
```

- .sdata = 0x20000
- Ig = 0x20000 + 0x400

What is GP ?

- ___global_pointer$
- defaul： .sdata + 0x800

How Linker Optimisation reduce the code size

```
1 lui  a0, %hi(Ig)
2 lw   a1, %lo(Ig)(a0)
```

```
1 lui  a0, 0x20000
2 lw   a1, 1024(a0)
```

- .sdata = 0x20000
- Ig = 0x20000 + 0x400

How Linker Optimisation reduce the code size

```
1 lui  a0, %hi(Ig)
2 lw   a1, %lo(Ig)(a0)
```

```
1 lui  a0, 0x20000
2 lw   a1, 1024(a0)
```

- .sdata = 0x20000
- Ig = 0x20000 + 0x400
- __global_pointer$ = .sdata + 0x800 = 0x20800

How Linker Optimisation reduce the code size

```
1 lui  a0, %hi(Ig)
2 lw   a1, %lo(Ig)(a0)
```

```
1 lui  a0, 0x20000
2 lw   a1, 1024(a0)
```

```
1 lw   a1, %lo(Ig)(a0)
```

```
1 lw   a1, -1024(gp)
```

- .sdata = 0x20000
- Ig = 0x20000 + 0x400
- __global_pointer$ = .sdata + 0x800 = 0x20800

How Linker Optimisation reduce the code size

```
1 lui  a0, %hi(Ig)
2 lw   a1, %lo(Ig)(a0)
```

```
1 lui  a0, 0x20000
2 lw   a1, 1024(a0)
```

```
1 lw   a1, %lo(Ig)(a0)
```

```
1 lw   a1, -1024(gp)
```

```
1 lui  a0, 0x21800
2 lw   a1, 0(Vg)
```

- .sdata = 0x20000
- Ig = 0x20000 + 0x400
- __global_pointer$ = .sdata + 0x800 = 0x20800

- Vg = 0x21800
- Vg - GP = 0x1000

Difference between ‘lw‘ and ‘lwgp‘

31:29	28:25	24:20	19:15	14:12	11:7	6 : 0	instruction
000	imm[8:2,10:9]		imm[15:11]	011	rd	0000111	LWGP

- lw rd, offset(rs1) -> lwgp rd, offset
- 12-bit/4KB -> 16-bit/64KB
- 4 bit alignment

```
1 lui  a0, %hi(Vg)
2 lw   a0, %lo(Vg)(a0)
```

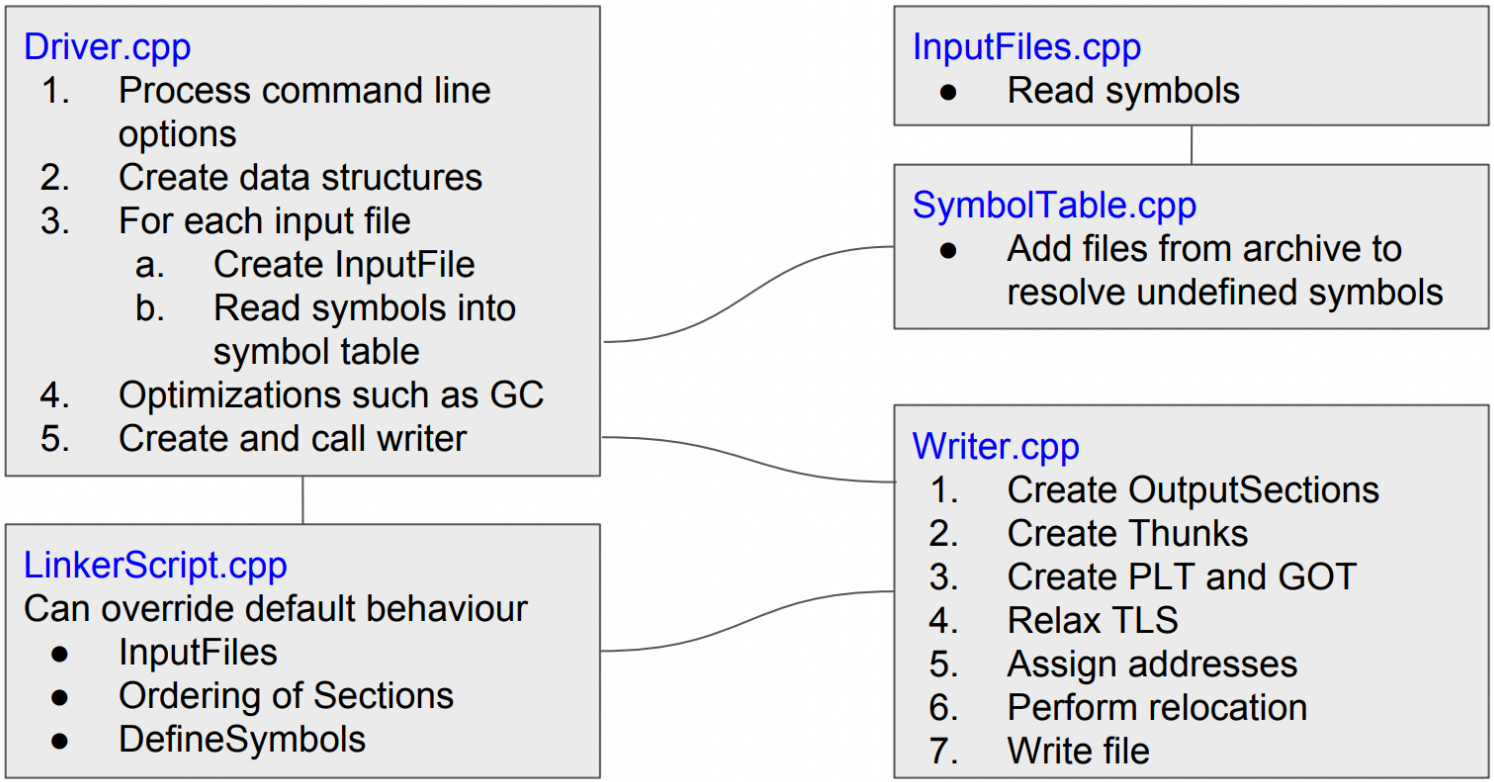
```
1 lui  a0, 0x21800
2 lw   a0, 0(a0)
```

- .sdata = 0x20000
- __global_pointer$ = .sdata + 0x800 = 0x20800
- Vg = 0x21800
- Vg - GP = 0x1000

```
1 lui  a0, %hi(Vg)
2 lw   a0, %lo(Vg)(a0)
```

```
1 lwgp   a0, 0x1000
```

- .sdata = 0x20000
- __global_pointer$ = .sdata + 0x800 = 0x20800
- Vg = 0x21800
- Vg - GP = 0x1000



- LLD supporting status of RISCV (BV1RQ4y127q5)
- Introduction of LLD (BV1Jg411j7Rd)

Define new relocation type

- llvm/include/llvm/BinaryFormat/ELFRelocs/RISCV.def
- github.com/riscv-non-isa/riscv-elf-psabi-doc

```
ELF_RELOC(R_RISCV_GPREL_ZCE_LWGP,     59)
ELF_RELOC(R_RISCV_GPREL_ZCE_SWGP,     60)
ELF_RELOC(R_RISCV_GPREL_ZCE_LDGP,     61)
ELF_RELOC(R_RISCV_GPREL_ZCE_SDGP,     62)
```

Mapping RelExpr

- R_RISCV_GPREL

```
case R_RISCV_GPREL: {
  if (!ElfSym::riscvGlobalPointer)
    llvm_unreachable(
        "Cannot compute R_RISCV_GPREL if __global_pointer$ is not set");

  return sym.getVA(a) - ElfSym::riscvGlobalPointer->getVA();
}
```

Mapping RelExpr

- RISCV::getRelExpr
- lld/ELF/Arch/RISCV.cpp

```
274   case R_RISCV_GPREL_I:    // lui、lw
275   case R_RISCV_GPREL_S:
276   case R_RISCV_GPREL_ZCE_LWGP:  // lwgp
277   case R_RISCV_GPREL_ZCE_SWGP:
278   case R_RISCV_GPREL_ZCE_LDGP:
279   case R_RISCV_GPREL_ZCE_SDGP:
280     return R_RISCV_GPREL;
```

- relaxHi20Lo12
- lld/ELF/Arch/RISCV.cpp

```
// lui a0, %hi(foo)(a0)
if (rel.type == R_RISCV_HI20) {
  if(isShiftedInt<14,2>(offset)){
    addDeleteRange(deleteRanges, rel.offset, 4);
    rel.type = R_RISCV_NONE;
    rel.expr = R_NONE;
    return true;
  }
}
// lw a0, %lo(foo)(a0)
else {
  unsigned rd = (inst & 0x00000fe0) >> 7;
  if(isShiftedInt<14,2>(offset)){
    newInst = (0x3007 | rd << 7); // lwgp rs, 0(gp)
    rel.type = R_RISCV_GPREL_ZCE_LWGP;
}
```

- RISCV::relocate(uint8_t *loc, const Relocation rel, uint64_t val)
- lld/ELF/Arch/RISCV.cpp

```
483   case R_RISCV_GPREL_ZCE_LWGP: {
484     unsigned inst = read32le(loc);
485
486     uint64_t imm8_2 = (val >> 2) & 0x7f;
487     uint64_t imm10_9 = (val >> 9) & 0x3;
488     uint64_t imm15_11 = (val >> 11) & 0x1f;
489
490     write32le(loc,
491               (inst | imm8_2 << 22 |
492               imm10_9 << 20 | imm15_11 << 15)); // lwgp rs, val(gp)
493     return;
494   }
```

Reference

plctlab/llvm-project - riscv-zce-extension

riscv-non-isa/riscv-elf-psabi-doc GitHub. (Accessed: 21 December 2021)

RISC-V gp global pointer register description - Wahahahehehe. (Accessed: 17 December 2021)

Smith, P. 'How to add a new target to LLD', p. 26

Thanks!