

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines

## [PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg11998.html>

---

- *From:* Avi Kivity <[avi@xxxxxxxxxxxxx](mailto:avi@xxxxxxxxxxxxx)>
  - *Date:* Mon, 31 Mar 2008 17:37:20 +0300
- 

From: Harvey Harrison <[harvey.harrison@xxxxxxxxxx](mailto:harvey.harrison@xxxxxxxxxx)>

Change jmp\_rel() to a function as well.

Signed-off-by: Harvey Harrison <[harvey.harrison@xxxxxxxxxx](mailto:harvey.harrison@xxxxxxxxxx)>

Signed-off-by: Avi Kivity <[avi@xxxxxxxxxxxxx](mailto:avi@xxxxxxxxxxxxx)>

---

arch/x86/kvm/x86\_emulate.c | 56 ++++++-----

1 files changed, 26 insertions(+), 30 deletions(-)

diff --git a/arch/x86/kvm/x86\_emulate.c b/arch/x86/kvm/x86\_emulate.c

index 008db4d..cacdcf5 100644

--- a/arch/x86/kvm/x86\_emulate.c

+++ b/arch/x86/kvm/x86\_emulate.c

@@ -501,23 +501,19 @@ register\_address(struct decode\_cache \*c, unsigned long base, unsigned long reg)

return base + address\_mask(c, reg);

}

+#define register\_address\_increment(reg, inc) \

- do { \

- /\* signed type ensures sign extension to long \*/ \

- int \_inc = (inc); \

- if (c->ad\_bytes == sizeof(unsigned long)) \

- (reg) += \_inc; \

- else \

- (reg) = ((reg) & \

- ~ad\_mask(c)) | \

- (((reg) + \_inc) & \

- ad\_mask(c)); \

- } while (0)

+static inline void

+register\_address\_increment(struct decode\_cache \*c, unsigned long \*reg, int inc)

+{

+ if (c->ad\_bytes == sizeof(unsigned long))

+ \*reg += inc;

+ else

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines 1

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines

```

+ *reg = (*reg & ~ad_mask(c)) | ((*reg + inc) & ad_mask(c));
+}

-#define JMP_REL(rel) \
- do { \
- register_address_increment(c->eip, rel); \
- } while (0)
+static inline void jmp_rel(struct decode_cache *c, int rel)
+{
+ register_address_increment(c, &c->eip, rel);
+}

static int do_fetch_insn_byte(struct x86_emulate_ctxt *ctxt,
struct x86_emulate_ops *ops,
@@ -1065,7 +1061,7 @@ static inline void emulate_push(struct x86_emulate_ctxt *ctxt)
c->dst.type = OP_MEM;
c->dst.bytes = c->op_bytes;
c->dst.val = c->src.val;
- register_address_increment(c->regs[VCPU_REGS_RSP], -c->op_bytes);
+ register_address_increment(c, &c->regs[VCPU_REGS_RSP], -c->op_bytes);
c->dst.ptr = (void *) register_address(c, ctxt->ss_base,
c->regs[VCPU_REGS_RSP]);
}
@@ -1082,7 +1078,7 @@ static inline int emulate_grpl1a(struct x86_emulate_ctxt *ctxt,
if (rc != 0)
return rc;

- register_address_increment(c->regs[VCPU_REGS_RSP], c->dst.bytes);
+ register_address_increment(c, &c->regs[VCPU_REGS_RSP], c->dst.bytes);

return 0;
}
@@ -1395,7 +1391,7 @@ special_insn:
c->dst.type = OP_MEM;
c->dst.bytes = c->op_bytes;
c->dst.val = c->src.val;
- register_address_increment(c->regs[VCPU_REGS_RSP],
+ register_address_increment(c, &c->regs[VCPU_REGS_RSP],
- c->op_bytes);
c->dst.ptr = (void *) register_address(
c, ctxt->ss_base, c->regs[VCPU_REGS_RSP]);
@@ -1407,7 +1403,7 @@ special_insn:
c->op_bytes, ctxt->vcpu) != 0)
goto done;

- register_address_increment(c->regs[VCPU_REGS_RSP],
+ register_address_increment(c, &c->regs[VCPU_REGS_RSP],
c->op_bytes);
c->dst.type = OP_NONE; /* Disable writeback. */
break;
@@ -1459,7 +1455,7 @@ special_insn:

```

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines2

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines

```
int rel = insn_fetch(s8, 1, c->eip);

if (test_cc(c->b, ctxt->eflags))
- JMP_REL(rel);
+ jmp_rel(c, rel);
break;
}
case 0x80 ... 0x83: /* Grp1 */
@@ -1545,10 +1541,10 @@ special_insn:
&c->dst.val,
c->dst.bytes, ctxt->vcpu)) != 0)
goto done;
- register_address_increment(c->regs[VCPU_REGS_RSI],
+ register_address_increment(c, &c->regs[VCPU_REGS_RSI],
(ctxt->eflags & EFLG_DF) ? -c->dst.bytes
: c->dst.bytes);
- register_address_increment(c->regs[VCPU_REGS_RDI],
+ register_address_increment(c, &c->regs[VCPU_REGS_RDI],
(ctxt->eflags & EFLG_DF) ? -c->dst.bytes
: c->dst.bytes);
break;
@@ -1580,10 +1576,10 @@ special_insn:

emulate_2op_SrcV("cmp", c->src, c->dst, ctxt->eflags);

- register_address_increment(c->regs[VCPU_REGS_RSI],
+ register_address_increment(c, &c->regs[VCPU_REGS_RSI],
(ctxt->eflags & EFLG_DF) ? -c->src.bytes
: c->src.bytes);
- register_address_increment(c->regs[VCPU_REGS_RDI],
+ register_address_increment(c, &c->regs[VCPU_REGS_RDI],
(ctxt->eflags & EFLG_DF) ? -c->dst.bytes
: c->dst.bytes);

@@ -1595,7 +1591,7 @@ special_insn:
ctxt->es_base,
c->regs[VCPU_REGS_RDI]);
c->dst.val = c->regs[VCPU_REGS_RAX];
- register_address_increment(c->regs[VCPU_REGS_RDI],
+ register_address_increment(c, &c->regs[VCPU_REGS_RDI],
(ctxt->eflags & EFLG_DF) ? -c->dst.bytes
: c->dst.bytes);
break;
@@ -1611,7 +1607,7 @@ special_insn:
c->dst.bytes,
ctxt->vcpu)) != 0)
goto done;
- register_address_increment(c->regs[VCPU_REGS_RSI],
+ register_address_increment(c, &c->regs[VCPU_REGS_RSI],
(ctxt->eflags & EFLG_DF) ? -c->dst.bytes
: c->dst.bytes);
```

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines3

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines

```
break;
@@ -1650,14 +1646,14 @@ special_insn:
goto cannot_emulate;
}
c->src.val = (unsigned long) c->eip;
- JMP_REL(rel);
+ jmp_rel(c, rel);
c->op_bytes = c->ad_bytes;
emulate_push(ctxt);
break;
}
case 0xe9: /* jmp rel */
case 0xeb: /* jmp rel short */
- JMP_REL(c->src.val);
+ jmp_rel(c, c->src.val);
c->dst.type = OP_NONE; /* Disable writeback. */
break;
case 0xf4: /* hlt */
@@ -1857,7 +1853,7 @@ twobyte_insn:
goto cannot_emulate;
}
if (test_cc(c->b, ctxt->eflags))
- JMP_REL(rel);
+ jmp_rel(c, rel);
c->dst.type = OP_NONE;
break;
}
--
1.5.4.5
```

---

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in  
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
Please read the FAQ at <http://www.tux.org/lkml/>

[PATCH 36/40] KVM: x86 emulator: make register\_address\_increment and JMP\_REL static inlines4