

## [PATCH 05/40] KVM: x86 emulator: Group decoding for group 3

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-03/msg12021.html>

---

- *From:* Avi Kivity <[avi@xxxxxxxxxxxxx](mailto:avi@xxxxxxxxxxxxx)>
  - *Date:* Mon, 31 Mar 2008 17:36:49 +0300
- 

This adds group decoding support for opcodes 0xf6, 0xf7 (group 3).

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

---

arch/x86/kvm/x86\_emulate.c | 34 ++++++-----  
1 files changed, 10 insertions(+), 24 deletions(-)

diff --git a/arch/x86/kvm/x86\_emulate.c b/arch/x86/kvm/x86\_emulate.c  
index cf1ce7c..52e65ae 100644

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

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

@@ -70,7 +70,7 @@

#define GroupMask 0xff /\* Group number stored in bits 0:7 \*/

enum {

- Group1A,

+ Group1A, Group3\_Byte, Group3,

};

static u16 opcode\_table[256] = {

@@ -171,8 +171,7 @@ static u16 opcode\_table[256] = {

0, 0, 0, 0,

/\* 0xF0 - 0xF7 \*/

0, 0, 0, 0,

- ImplicitOps, ImplicitOps,

- ByteOp | DstMem | SrcNone | ModRM, DstMem | SrcNone | ModRM,

+ ImplicitOps, ImplicitOps, Group | Group3\_Byte, Group | Group3,

/\* 0xF8 - 0xFF \*/

ImplicitOps, 0, ImplicitOps, ImplicitOps,

0, 0, ByteOp | DstMem | SrcNone | ModRM, DstMem | SrcNone | ModRM

@@ -239,6 +238,14 @@ static u16 twobyte\_table[256] = {

static u16 group\_table[] = {

[Group1A\*8] =

DstMem | SrcNone | ModRM | Mov | Stack, 0, 0, 0, 0, 0, 0, 0,

+ [Group3\_Byte\*8] =

+ ByteOp | SrcImm | DstMem | ModRM, 0,

+ ByteOp | DstMem | SrcNone | ModRM, ByteOp | DstMem | SrcNone | ModRM,

+ 0, 0, 0, 0,

[PATCH 05/40] KVM: x86 emulator: Group decoding for group 3

```
+ [Group3*8] =  
+ DstMem | SrcImm | ModRM | SrcImm, 0,  
+ DstMem | SrcNone | ModRM, ByteOp | DstMem | SrcNone | ModRM,  
+ 0, 0, 0, 0,  
};
```

```
static u16 group2_table[] = {  
@@ -1070,26 +1077,6 @@ static inline int emulate_grp3(struct x86_emulate_ctxt *ctxt,  
  
switch (c->modrm_reg) {  
case 0 ... 1: /* test */  
- /*  
- * Special case in Grp3: test has an immediate  
- * source operand.  
- */  
- c->src.type = OP_IMM;  
- c->src.ptr = (unsigned long *)c->eip;  
- c->src.bytes = (c->d & ByteOp) ? 1 : c->op_bytes;  
- if (c->src.bytes == 8)  
- c->src.bytes = 4;  
- switch (c->src.bytes) {  
- case 1:  
- c->src.val = insn_fetch(s8, 1, c->eip);  
- break;  
- case 2:  
- c->src.val = insn_fetch(s16, 2, c->eip);  
- break;  
- case 4:  
- c->src.val = insn_fetch(s32, 4, c->eip);  
- break;  
- }  
emulate_2op_SrcV("test", c->src, c->dst, ctxt->eflags);  
break;  
case 2: /* not */  
@@ -1103,7 +1090,6 @@ static inline int emulate_grp3(struct x86_emulate_ctxt *ctxt,  
rc = X86EMUL_UNHANDLEABLE;  
break;  
}  
-done:  
return rc;  
}
```

---  
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/>