南京航空航天大学 计算机科学与技术系学院 计算机组成原理 课程实验

学号：161630220

姓名：赵维康

PA3- 穿越时空的旅程： 异常控制流

在进行本 PA 前,请在工程目录下执行以下命令进行分支整理,否则将影响你的成绩:

git commit --allow-empty -am "before starting pa3"

git checkout master

git merge pa2

git checkout -b pa3

如图

```
zhaoweikang@zhaoweikang:~/ics2017$ sudo git commit --allow-empty -am "before sta
rting pa3"
[sudo] zhaoweikang 的密码：
[pa2 07761d6] before starting pa3
 15 files changed, 580 insertions(+), 120 deletions(-)
zhaoweikang@zhaoweikang:~/ics2017$ sudo git checkout master
切换到分支 'master'
您的分支领先 'origin/2017' 共 58 个提交。
  （使用 "git push" 来发布您的本地提交）
zhaoweikang@zhaoweikang:~/ics2017$ sudo git merge pa2
更新 cc11412..07761d6
Fast-forward
 nemu/include/common.h                      |   4 +-
 nemu/include/cpu/reg.h                     |  19 ++-
 nemu/include/cpu/rtl.h                     |  45 ++++--
 nemu/src/cpu/decode/decode.c               |  17 +-
 nemu/src/cpu/exec/all-instr.h              |  93 ++++++++++
 nemu/src/cpu/exec/arith.c                  |  93 ++++++++++-
 nemu/src/cpu/exec/cc.c                     |  15 +-
 nemu/src/cpu/exec/control.c                |  14 +-
 nemu/src/cpu/exec/data-mov.c               |  45 ++++--
 nemu/src/cpu/exec/exec.c                   | 250 ++++++++++++++++++++++++--------
 nemu/src/cpu/exec/logic.c                  |  72 +++++++--
 nemu/src/cpu/exec/system.c                 |   4 +-
 nemu/src/memory/memory.c                   |   9 +-
 nemu/src/monitor/diff-test/diff-test.c     |  18 ++-
 nemu/src/monitor/monitor.c                 |   2 +
 nexus-am/Makefile.check                    |   4 +-
 nexus-am/am/arch/x86-nemu/img/run          |   1 +
 nexus-am/am/arch/x86-nemu/src/ioe.c        |  26 +++-
 nexus-am/am/arch/x86-nemu/src/trm.c        |   2 +-
 nexus-am/tests/cputest/dummy-x86-nemu.txt |   0
 20 files changed, 606 insertions(+), 127 deletions(-)
 create mode 100644 nexus-am/tests/cputest/dummy-x86-nemu.txt
zhaoweikang@zhaoweikang:~/ics2017$ sudo git checkout -b pa3
```

```
切换到一个新分支 'pa3'
```

加载操作系统的第一个用户程序

首先进入 navy-apps/Makefile.check 文件，让 Navy-apps 项目上的程序默认编译到 x86 中：即注释掉 ISA ？=navtive，添加 ISA ？=x86，如图

```
//ISA ?= native
ISA ?= x86
ifeq ($(NAVY_HOME), )
  $(error Must set NAVY_HOME environment variable)
endif

$(shell mkdir -p $(NAVY_HOME)/fsimg/bin/ $(NAVY_HOME)/fsimg/dev/)
```

在 navy-apps/tests/dummy 下执行 make 命令，如图，warning 较多，故省去 warning

```
root@zhaoweikang:/home/zhaoweikang/ics2017/navy-apps/tests/dummy# make
make -C /home/zhaoweikang/ics2017/navy-apps/libs/libc
make[1]: Entering directory '/home/zhaoweikang/ics2017/navy-apps/libs/libc'
+ CC src/stdio/remove.c
```

```
+ CC src/signal/raise.c
+ CC src/signal/signal.c
+ AR /home/zhaoweikang/ics2017/navy-apps/libs/libc/build/libc-x86.a
make[1]: Leaving directory '/home/zhaoweikang/ics2017/navy-apps/libs/libc'
make -C /home/zhaoweikang/ics2017/navy-apps/libs/libos
make[1]: Entering directory '/home/zhaoweikang/ics2017/navy-apps/libs/libos'
+ CC src/nanos.c
+ AR /home/zhaoweikang/ics2017/navy-apps/libs/libos/build/libos-x86.a
make[1]: Leaving directory '/home/zhaoweikang/ics2017/navy-apps/libs/libos'
+ CC dummy.c
+ LD /home/zhaoweikang/ics2017/navy-apps/tests/dummy/build/dummy-x86
```

执行 make 后，在 navy-apps/tests/dummy/build/目录下生成 dummy 的可执行文件，如图

```
zhaoweikang@zhaoweikang:~/ics2017/navy-apps/tests/dummy/build$ ls
dummy-x86  x86
```

在 nanos-lite/ 目录下执行 make update 命令，如图

```
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make update
Building nanos-lite [x86-nemu]
objcopy -S --set-section-flags .bss=alloc,contents -O binary /home/zhaoweikang/i
cs2017/navy-apps/tests/dummy/build/dummy-x86 build/ramdisk.img
touch src/files.h
ln -sf /home/zhaoweikang/ics2017/navy-apps/libs/libos/src/syscall.h src/syscall.
h
```

然后进入 nanos-lite 目录下刚刚编译生成的 build 文件，就会发现由 nanos-lite/Makefile 生成的 ramdisk 的镜像文件 ramdisk.img，如图

```
zhaoweikang@zhaoweikang:~/ics2017/nanos-lite$ cd build/
zhaoweikang@zhaoweikang:~/ics2017/nanos-lite/build$ ls
ramdisk.img  x86-nemu
```

实现 loader

进入 nanos-lite/src/loader.c 中，直接调用 ramdisc_read()函数，读取 get_ramdisc_size()大小的数据到 DEFAULT_ENTRY 即可，再返回 DEFAULT_ENTRY 即可，如图

```
uintptr_t loader(_Protect *as, const char *filename) {
  ramdisk_read(DEFAULT_ENTRY, 0, get_ramdisk_size());


      |


       return (uintptr_t)DEFAULT_ENTRY;
}
```

在 nanos-lite/下运行 dummy 程序，出现了艺术字 i386，查看 build 目录下的反汇编代码知，是 int 指令未实现。

准备 IDT

先进入，nemu/include/cpu/reg.h 文件，在 cpu 结构体中添加 IDTR 寄存器，同时添加 cs 段寄存器，以便在 QEMU 中进行 Differential testing，如图

```
  struct{
      rtlreg_t eax, ecx, edx, ebx, esp, ebp, esi, edi;
         };
       };
      vaddr_t eip;
      unsigned int cs;
       union{
            rtlreg_t eflags_init;
            struct{
                 unsigned int CF:1;
                 unsigned int ZF:1;
                 unsigned int SF:1;
                 unsigned int IF:1;
                 unsigned int OF:1;
                     };
               }eflags;
       struct {
                uint16_t limit;
                uint32_t base;
               }idtr;

} CPU_state;
```

查 i386 手册知，IDTR 有 the linear base address and limit values，即 idt 的首地址（16）和长度（32 位）
接下来，进入 nemu/src/monitor/monitor.c 文件的 restart（）函数中，将 cs 初始化为 8，将 EFLAGS 初始化为 2，如图

```
static inline void restart() {
  /* Set the initial instruction pointer. */
  cpu.eip = ENTRY_START;
        cpu.cs = 0x8;
        cpu.eflags.eflags_init =0x2;

#ifdef DIFF_TEST
  init_qemu_reg();
#endif
}
```

下面进入 nemu/src/cpu/intr.c 文件，实现 raise_intr()函数，如图

```
#include "cpu/exec.h"
#include "memory/mmu.h"

void raise_intr(uint8_t NO, vaddr_t ret_addr) {
  /* TODO: Trigger an interrupt/exception with ``NO''.
   * That is, use ``NO'' to index the IDT.
   */
  rtl_push((rtlreg_t *)&cpu.eflags);
  rtl_push((rtlreg_t *)&cpu.cs);
  rtl_push((rtlreg_t *)&ret_addr);
  uint32_t idtr_base = cpu.idtr.base;
  uint32_t eip_low,eip_high,offset;
  eip_low = vaddr_read(idtr_base + NO * 8,4) & 0x0000ffff;
  eip_high = vaddr_read(idtr_base + NO * 8 + 4,4) & 0xffff0000;
  offset = eip_low | eip_high;
  decoding.jmp_eip = offset;
  decoding.is_jmp = true;
}
```

执行 make 及 make run 命令，如图

```
zhaoweikang@zhaoweikang:~/ics2017/nemu$ sudo make
[sudo] zhaoweikang 的密码：
+ CC src/memory/memory.c
+ CC src/monitor/cpu-exec.c
+ CC src/monitor/monitor.c
+ CC src/monitor/debug/expr.c
+ CC src/monitor/debug/watchpoint.c
+ CC src/monitor/debug/ui.c
+ CC src/monitor/diff-test/diff-test.c
+ CC src/cpu/decode/modrm.c
+ CC src/cpu/decode/decode.c
+ CC src/cpu/reg.c
+ CC src/cpu/exec/cc.c
+ CC src/cpu/exec/arith.c
+ CC src/cpu/exec/control.c
+ CC src/cpu/exec/prefix.c
+ CC src/cpu/exec/logic.c
+ CC src/cpu/exec/system.c
+ CC src/cpu/exec/data-mov.c
```

```
+ CC src/cpu/exec/exec.c
+ CC src/cpu/exec/special.c
+ CC src/cpu/intr.c
fatal: ..：'..' 在仓库之外
Makefile:41: recipe for target 'build/nemu' failed
make: [build/nemu] Error 128 (ignored)
+ LD build/nemu
zhaoweikang@zhaoweikang:~/ics2017/nemu$ sudo make run
```

```
./build/nemu -l ./build/nemu-log.txt
[src/monitor/monitor.c,47,load_default_img] No image is given. Use the default b
uild-in image.
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x00100026
```

说明实现没有错误

实现 raise_intr()函数，即实现中断机制，可分为以下几步来实现：

1.依次将 EFLAGS，CS，EIP 寄存器的值压入堆栈

2.从 IDTR 中读出 IDT 的首地址

3.根据异常(中断)号在 IDT 中进行索引,找到一个门描述符

4.将门描述符中的 offset 域组合成目标地址

5.跳转到目标地址

为了实现 int、lidt 指令，先在 nemu/src/exec/all-instr.h 文件中，对 int、lidt 的执行函数进行声明，同时将 iret 指令也一并声明，如图

```
make_EHelper(in);

make_EHelper(out);

make_EHelper(int);

make_EHelper(lidt);

make_EHelper(iret);
```

下面进入 nemu/src/exec/system.c 文件，实现 int、lidt 的执行函数，如图

```
make_EHelper(int) {
  raise_intr(id_dest->val,decoding.seq_eip);

  print_asm("int %s", id_dest->str);

#ifdef DIFF_TEST
  diff_test_skip_nemu();
#endif
}
```

int：查手册知，只需实现 int 的 REAL-ADDRESS-MODE 即可，而 raise_intr()的实现就是基于中断机制之一原理，直接调用即可。而 raise_intr()的两个参数，中断 NO 在目的操作数中，中断返回指令 iret 就是下一条指令，即 decoding.seq_eip。

```
make_EHelper(lidt) {
  cpu.idtr.limit = vaddr_read(id_dest->addr,2);
  if(decoding.is_operand_size_16) {
  cpu.idtr.base = vaddr_read(id_dest->addr + 2,4) & 0x00ffffff;
  }
  else {
  cpu.idtr.base = vaddr_read(id_dest->addr + 2,4);
  }

  print_asm_template1(lidt);
}
```

lidt:查手册知，idtr.limit 是 m16，当操作数是 16 位（两字节）时，idtr.base（首地址）的高八位舍去，否则，idtr.base（首地址）的高八位保留。

下面进入 nemu/src/exec/exec.c 文件，填写 opcode_table[]译码表，如图

int

```
  /* 0xcc */    EMPTY, IDEXW(I, int, 1), EMPTY, EMPTY,
```

lidt 框架已经实现，只需在 make_group()宏中加入 lidt 即可

```
make_group(gp7,
    EMPTY, EMPTY, EMPTY, EX(lidt),

    EMPTY, EMPTY, EMPTY, EMPTY)
```

执行 make 及 make run 命令，如图

```
root@zhaoweikang:/home/zhaoweikang/ics2017/nemu# make
+ CC src/cpu/exec/system.c
+ CC src/cpu/exec/exec.c

+ LD build/nemu
root@zhaoweikang:/home/zhaoweikang/ics2017/nemu# make run
```

```
./build/nemu -l ./build/nemu-log.txt
[src/monitor/monitor.c,47,load_default_img] No image is given. Use the default b
uild-in image.
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x00100026
```

说明实现没有错误

后在 nanos-lite/src/main.c 中定义宏 HAS_ASYE（去掉注释即可），如图

```
#include "common.h"

/* Uncomment these macros to enable corresponding functionality. */
#define HAS_ASYE
//#define HAS_PTE
```

下面切换至 nanos-lite 目录下，执行 make update 更新 ramdisk 的内容，再执行 make run 命令，运行 dummy 程序，如图

```
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make update
Building nanos-lite [x86-nemu]
objcopy -S --set-section-flags .bss=alloc,contents -O binary /home/zhaoweikang/i
cs2017/navy-apps/tests/dummy/build/dummy-x86 build/ramdisk.img
touch src/files.h
```

```
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make run
Building nanos-lite [x86-nemu]
+ CC src/device.c
+ CC src/fs.c
+ CC src/loader.c
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am'
make[2]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/am'
Building am [x86-nemu]
make[2]: Nothing to be done for 'archive'.
make[2]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am/am'
make[1]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am'
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/libs/klib'
make[1]: *** 没有指明目标并且找不到 makefile。 停止。
```

```
./build/nemu -l /home/zhaoweikang/ics2017/nanos-lite/build/nemu-log.txt /home/zh
aoweikang/ics2017/nanos-lite/build/nanos-lite-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/zhaoweikang/ics2017/nanos
-lite/build/nanos-lite-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
 [src/main.c,19,main] 'Hello World!' from Nanos-lite
[src/main.c,20,main] Build time: 22:54:51, May 12 2018
[src/ramdisk.c,26,init_ramdisk] ramdisk info: start = 0x100ef4, end = 0x106230,
size = 21308 bytes
[src/main.c,27,main] Initializing interrupt/exception handler...
invalid opcode(eip = 0x00100b29): 60 54 e8 fc fe ff ff 83 ...

There are two cases which will trigger this unexpected exception:
1. The instruction at eip = 0x00100b29 is not implemented.
2. Something is implemented incorrectly.
Find this eip(0x00100b29) in the disassembling result to distinguish which case
it is.
```

```
If it is the first case, see
 _ _____   ___    __    __ __                          _
(_)___ \ / _ \  / /   |  \/  |                        | |
  _  __) | (_) |/ /_   | \  / | __ _ _ __   _   _  __ _| |
 | ||__ < > _ <| '_ \  | |\/| |/ _` | '_ \| | | |/ _` | |
 | |___) | (_) | (_) | | |  | | (_| | | | | |_| | (_| | |
 |_|____/ \___/ \___/  |_|  |_|\__,_|_| |_|\__,_|\__,_|_|

for more details.

If it is the second case, remember:
* The machine is always right!
* Every line of untested code is always wrong!

(nemu) q
```

查看 **nanos-lite/build/x86-nemu.txt** 反汇编文件里对应的指令，发现是 **pusha** 指令未实现，说明中断机制实现正确，如图

```
00100b29 <asm_trap>:
  100b29:       60                      pusha
  100b2a:       54                      push   %esp
  100b2b:       e8 fc fe ff ff          call   100a2c <irq_handle>
  100b30:       83 c4 04                add    $0x4,%esp
  100b33:       61                      popa
  100b34:       83 c4 08                add    $0x8,%esp
  100b37:       cf                      iret
```

保存现场

思考题：对比异常与函数调用

我们知道进行函数调用的时候也需要保存调用者的状态:返回地址,以及调用约定(calling convention)中需要调用者保存的寄存器。而进行异常处理之前却要保存更多的信息。尝试对比它们，并思考两者保存信息不同是什么原因造成的。

答：在异常处理之后，系统要返回异常处理之前的状态，就是所谓的保护现场，不然就无法回到异常处理之前的状态。

思考题：诡异的代码

trap.S 中有一行 pushl %esp 的代码, 乍看之下其行为十分诡异。你能结合前后的代码理解它的行为吗？Hint:不用想太多,其实都是你学过的知识。

答：就是将异常处理指令压栈的操作。

重新组织 TrapFrame 结构体

实现 pusha 指令，先进入 nemu/src/exec/all-instr.h 文件，对 pusha 指令的执行函数进行声明，如图

```
make_EHelper(push);

make_EHelper(pop);

make_EHelper(pusha);

make_EHelper(leave);
```

进入 nemu/src/cpu/exec/data-mov.c 文件，实现 pusha 的执行函数，如图

```
make_EHelper(pusha) {
  t0 = cpu.esp;
  rtl_push(&cpu.eax);
  rtl_push(&cpu.ecx);
  rtl_push(&cpu.edx);
  rtl_push(&cpu.ebx);
  rtl_push(&t0);
  rtl_push(&cpu.ebp);
  rtl_push(&cpu.esi);
  rtl_push(&cpu.edi);

  print_asm("pusha");
}
```

pusha:查手册知，首先将 esp 保存在临时变量 temp 中，然后依次将 eax、ecx、edx、ebx、esp（临时变量）、ebp、esi、edi 入栈。

进入 nemu/src/cpu/exec/exec.c 文件，填写 pusha 的 opcode_table,如图

```
  /* 0x60 */    EX(pusha), EMPTY, EMPTY, EMPTY,
```

然后执行 make 及 make run 命令，如图

```
zhaoweikang@zhaoweikang:~/ics2017/nemu$ sudo make
+ CC src/cpu/exec/data-mov.c
+ CC src/cpu/exec/exec.c
```

```
+ LD build/nemu
zhaoweikang@zhaoweikang:~/ics2017/nemu$ sudo make run
./build/nemu -l ./build/nemu-log.txt
[src/monitor/monitor.c,47,load_default_img] No image is given. Use the default b
uild-in image.
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x00100026
```

说明实现无误

然后 nexus-am/am/arch/x86nemu/include/arch.h 中，对_RegSet 结构体的成员重新排序，以符合 trap frame 的形成过程，如图

```
struct _RegSet {
  uintptr_t edi,esi,ebp,esp,ebx,edx,ecx,eax;
  //uintptr_t esi, ebx, eax, eip, edx, error_code, eflags, ecx, cs, esp, edi, ebp;
  int       irq;
  uintptr_t error_code,eip,cs,eflags;
};
```

重新在 Nanos-lite 上运行 dummy 程序，在 nanos-lite/src/irq.c 中的 do_event() 函数中触发了 BAD TRAP，如图

```
root@zhaoweikang:/home/zhaoweikang/ics2017# cd nanos-lite/
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make update
Building nanos-lite [x86-nemu]
objcopy -S --set-section-flags .bss=alloc,contents -O binary /home/zhaoweikang/i
cs2017/navy-apps/tests/dummy/build/dummy-x86 build/ramdisk.img
touch src/files.h
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make run
Building nanos-lite [x86-nemu]
+ AS src/initrd.S
+ CC src/fs.c
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am'
make[2]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/am'
Building am [x86-nemu]
make[2]: Nothing to be done for 'archive'.
make[2]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am/am'
make[1]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am'
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/libs/klib'
make[1]: *** 没有指明目标并且找不到 makefile。 停止。
```

```
./build/nemu -l /home/zhaoweikang/ics2017/nanos-lite/build/nemu-log.txt /home/zh
aoweikang/ics2017/nanos-lite/build/nanos-lite-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/zhaoweikang/ics2017/nanos
-lite/build/nanos-lite-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
 [src/main.c,19,main] 'Hello World!' from Nanos-lite
[src/main.c,20,main] Build time: 13:29:46, May 13 2018
[src/ramdisk.c,26,init_ramdisk] ramdisk info: start = 0x100ef4, end = 0x106230,
size = 21308 bytes
[src/main.c,27,main] Initializing interrupt/exception handler...
[src/irq.c,5,do_event] system panic: Unhandled event ID = 8
nemu: HIT BAD TRAP at eip = 0x00100032
```

恢复现场

实现系统调用

进入 nanos-lite/src/irq.c 文件，在 do_event()中识别出系统调用事件 _EVENT_SYSCALL ，然后调用 do_syscall()，如图

```
#include "common.h"
 extern _RegSet* do_syscall(_RegSet* r);
 static _RegSet* do_event(_Event e, _RegSet* r) {
  switch (e.event) {
    case _EVENT_SYSCALL:
                        return do_syscall(r);
    default: panic("Unhandled event ID = %d", e.event);
  }

  return NULL;
}
```

然后在 nanos-lite 目录下执行 make update 及 make run 命令，触发了一个号码为 0 的 SYS_none 系统调用，如图

```
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make update
Building nanos-lite [x86-nemu]
objcopy -S --set-section-flags .bss=alloc,contents -O binary /home/zhaoweikang/i
cs2017/navy-apps/tests/dummy/build/dummy-x86 build/ramdisk.img
touch src/files.h
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make run
Building nanos-lite [x86-nemu]
+ AS src/initrd.S
+ CC src/irq.c
+ CC src/fs.c
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am'
make[2]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/am'
Building am [x86-nemu]
make[2]: Nothing to be done for 'archive'.
make[2]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am/am'
make[1]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am'
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/libs/klib'
make[1]: *** 没有指明目标并且找不到 makefile。 停止。

./build/nemu -l /home/zhaoweikang/ics2017/nanos-lite/build/nemu-log.txt /home/zh
aoweikang/ics2017/nanos-lite/build/nanos-lite-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/zhaoweikang/ics2017/nanos
-lite/build/nanos-lite-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
 [src/main.c,19,main] 'Hello World!' from Nanos-lite
[src/main.c,20,main] Build time: 13:29:46, May 13 2018
[src/ramdisk.c,26,init_ramdisk] ramdisk info: start = 0x100f9c, end = 0x1062d8,
size = 21308 bytes
[src/main.c,27,main] Initializing interrupt/exception handler...
[src/syscall.c,9,do_syscall] system panic: Unhandled syscall ID = 0
nemu: HIT BAD TRAP at eip = 0x00100032
```

下面进入 nexus-am/am/arch/x86-nemu/include/arch.h 中实现正确的 SYSCALL_ARGx()宏，让它们从作为参数的现场 reg 中获得正确的系统调用参数寄存器，如图

```
#define SYSCALL_ARG1(r) r->eax
#define SYSCALL_ARG2(r) r->ebx
#define SYSCALL_ARG3(r) r->ecx
#define SYSCALL_ARG4(r) r->edx
```

只要将系统调用参数 0 相应的改为 eax、ebx、ecx、edx 即可。

下面进入 nanos-lite/src/syscall.c 中添加 SYS_none 系统调用（这个系统调用什么都不用做,直接返回 1），设置系统调用的返回值（系统调用的返回值存放在系统调用号所在的寄存器中，通过 SYSCALL_ARG1()来进行设置），如图

```
#include "common.h"
#include "syscall.h"

_RegSet* do_syscall(_RegSet *r) {
  uintptr_t a[4],result = -1;
  a[0] = SYSCALL_ARG1(r);

  switch (a[0]) {
    case SYS_none:
                  result = 1;
                  break;
    default: panic("Unhandled syscall ID = %d", a[0]);
  }
  SYSCALL_ARG1(r) = result;
  return NULL;
}
```

下面实现 popa 和 iret 指令。
进入 nemu/src/cpu/exec/all-instr.h 文件，对 popa 的执行函数进行声明，iret 前面已经声明过，如图

```
make_EHelper(push);

make_EHelper(pop);

make_EHelper(pusha);

make_EHelper(popa);

make_EHelper(leave);
```

下面进入 nemu/src/cpu/exec/data-mov.c 实现 popa 的执行函数，如图

```
make_EHelper(popa) {
  rtl_pop(&cpu.edi);
  rtl_pop(&cpu.esi);
  rtl_pop(&cpu.ebp);
  rtl_pop(&t0);
  rtl_pop(&cpu.ebx);
  rtl_pop(&cpu.edx);
  rtl_pop(&cpu.ecx);
  rtl_pop(&cpu.eax);

  print_asm("popa");
}
```

popa:此指令与 pusha 刚好相反，最后入栈的先出栈，所以 edi 先出栈。需要注意的是，esp 并没有出栈，出栈的只是它的一个副本。

进入 nemu/src/cpu/exec/system.c 实现 iret 的执行函数，如图

```
make_EHelper(iret) {
  rtl_pop(&decoding.jmp_eip);
  rtl_pop(&cpu.cs);
  rtl_pop(&cpu.eflags.eflags_init);
  decoding.is_jmp = 1;

  print_asm("iret");
}
```

iret:此指令就是将 eip、cs、eflags 出栈。

下面进入 nemu/src/cpu/exec/exec.c 中，填写 popa、iret 的 opcode_table 表，如图

iret：

```
  /* 0xcc */    EMPTY, IDEXW(I, int, 1), EMPTY, EX(iret),
```

popa:

```
  /* 0x60 */    EX(pusha),  EX(popa), EMPTY, EMPTY,
```

然后执行 make 及 make run 命令，如图

```
zhaoweikang@zhaoweikang:~/ics2017/nemu$ sudo make
+ CC src/cpu/exec/system.c
+ CC src/cpu/exec/data-mov.c
+ CC src/cpu/exec/exec.c
+ LD build/nemu
zhaoweikang@zhaoweikang:~/ics2017/nemu$ sudo make run
./build/nemu -l ./build/nemu-log.txt
[src/monitor/monitor.c,47,load_default_img] No image is given. Use the default b
uild-in image.
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
nemu: HIT GOOD TRAP at eip = 0x00100026
```

说明实现没有错误

在 nanos-lite/下重新运行 dummy 程序，如图

```
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make update
Building nanos-lite [x86-nemu]
objcopy -S --set-section-flags .bss=alloc,contents -O binary /home/zhaoweikang/i
cs2017/navy-apps/tests/dummy/build/dummy-x86 build/ramdisk.img
touch src/files.h
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make run
Building nanos-lite [x86-nemu]
+ CC src/mm.c
+ AS src/initrd.S
+ CC src/ramdisk.c
+ CC src/syscall.c
+ CC src/proc.c
+ CC src/irq.c
+ CC src/main.c
+ CC src/device.c
+ CC src/fs.c
+ CC src/loader.c
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am'
make[2]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/am'
Building am [x86-nemu]
+ CC arch/x86-nemu/src/trm.c
+ CC arch/x86-nemu/src/pte.c
+ CC arch/x86-nemu/src/ioe.c
```

```
+ CC arch/x86-nemu/src/asye.c
+ AR /home/zhaoweikang/ics2017/nexus-am/am/build/am-x86-nemu.a
make[2]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am/am'
make[1]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am'
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/libs/klib'
make[1]: *** 没有指明目标并且找不到 makefile。 停止。
```

```
./build/nemu -l /home/zhaoweikang/ics2017/nanos-lite/build/nemu-log.txt /home/zh
aoweikang/ics2017/nanos-lite/build/nanos-lite-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/zhaoweikang/ics2017/nanos
-lite/build/nanos-lite-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
 [src/main.c,19,main] 'Hello World!' from Nanos-lite
[src/main.c,20,main] Build time: 17:18:58, May 13 2018
[src/ramdisk.c,26,init_ramdisk] ramdisk info: start = 0x100fc8, end = 0x106304,
size = 21308 bytes
[src/main.c,27,main] Initializing interrupt/exception handler...
[src/syscall.c,12,do_syscall] system panic: Unhandled syscall ID = 4
nemu: HIT BAD TRAP at eip = 0x00100032
```

dummy 程序又触发了一个号码为 4 的系统调用（即 SYS_exit 的系统调用）

下面进入 nanos-lite/src/syscall.c 中，实现SYS_exit 系统调用,它会接收一个退出状态的参数,用这个参数调用_halt() 即可，如图

```
_RegSet* do_syscall(_RegSet *r) {
  uintptr_t a[4],result = -1;
  a[0] = SYSCALL_ARG1(r);
  a[1] = SYSCALL_ARG2(r);
  switch (a[0]) {
    case SYS_none:
                  result = 1;
                  break;
    case SYS_exit:
                  _halt(a[1]);
                  break;
    default: panic("Unhandled syscall ID = %d", a[0]);
  }
  SYSCALL_ARG1(r) = result;
  return NULL;
}
```

在 nanos-lite/下再次运行 dummy 程序，如图

```
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make update
Building nanos-lite [x86-nemu]
objcopy -S --set-section-flags .bss=alloc,contents -O binary /home/zhaoweikang/i
cs2017/navy-apps/tests/dummy/build/dummy-x86 build/ramdisk.img
touch src/files.h
root@zhaoweikang:/home/zhaoweikang/ics2017/nanos-lite# make run
Building nanos-lite [x86-nemu]
+ AS src/initrd.S
+ CC src/syscall.c
+ CC src/fs.c
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am'
make[2]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/am'
Building am [x86-nemu]
make[2]: Nothing to be done for 'archive'.
make[2]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am/am'
make[1]: Leaving directory '/home/zhaoweikang/ics2017/nexus-am'
make[1]: Entering directory '/home/zhaoweikang/ics2017/nexus-am/libs/klib'
make[1]: *** 没有指明目标并且找不到 makefile。 停止。
```

```
./build/nemu -l /home/zhaoweikang/ics2017/nanos-lite/build/nemu-log.txt /home/zh
aoweikang/ics2017/nanos-lite/build/nanos-lite-x86-nemu.bin
[src/monitor/monitor.c,65,load_img] The image is /home/zhaoweikang/ics2017/nanos
-lite/build/nanos-lite-x86-nemu.bin
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 20:44:23, May 12 2018
For help, type "help"
(nemu) c
 [src/main.c,19,main] 'Hello World!' from Nanos-lite
[src/main.c,20,main] Build time: 17:18:58, May 13 2018
[src/ramdisk.c,26,init_ramdisk] ramdisk info: start = 0x100fec, end = 0x106328,
size = 21308 bytes
[src/main.c,27,main] Initializing interrupt/exception handler...
nemu: HIT GOOD TRAP at eip = 0x00100032
```

输出了 GOODTRAP 的信息，说明成功实现上述功能。

git log 记录

```
zhaoweikang@zhaoweikang:~/ics2017$ sudo git status
位于分支 pa3
尚未暂存以备提交的变更：
  （使用 "git add/rm <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

        删除：      Makefile
        删除：      README.md
        修改：      nanos-lite/src/device.c
        修改：      nanos-lite/src/fs.c
        修改：      nanos-lite/src/irq.c
        修改：      nanos-lite/src/loader.c
        修改：      nanos-lite/src/main.c
        修改：      nanos-lite/src/mm.c
        修改：      nanos-lite/src/syscall.c
        修改：      nemu/include/cpu/reg.h
        修改：      nemu/src/cpu/exec/all-instr.h
        修改：      nemu/src/cpu/exec/data-mov.c
        修改：      nemu/src/cpu/exec/exec.c
        修改：      nemu/src/cpu/exec/system.c
        修改：      nemu/src/cpu/intr.c
        修改：      nexus-am/am/arch/x86-nemu/include/arch.h

未跟踪的文件：
  （使用 "git add <文件>..." 以包含要提交的内容）

        navy-apps/

修改尚未加入提交（使用 "git add" 和/或 "git commit -a"）
zhaoweikang@zhaoweikang:~/ics2017$ sudo git add .
zhaoweikang@zhaoweikang:~/ics2017$ sudo git commit --allow-empty
[pa3 8a85233] fix bug for pa3.1
 37 files changed, 463 insertions(+), 60 deletions(-)
 delete mode 100644 Makefile
 delete mode 100644 README.md
 create mode 100644 navy-apps/Makefile
 create mode 100644 navy-apps/README.md
 create mode 100644 navy-apps/apps/init/Makefile
 create mode 100644 navy-apps/apps/litenes/Makefile
 create mode 100644 navy-apps/apps/lua/Makefile
 create mode 100644 navy-apps/apps/nterm/Makefile
 create mode 100644 navy-apps/apps/nwm/Makefile
 create mode 100644 navy-apps/apps/pal/Makefile
 create mode 100644 navy-apps/apps/pal/README.md
 create mode 100644 navy-apps/libs/libc/Makefile
 create mode 100644 navy-apps/libs/libc/README.md
 create mode 100644 navy-apps/libs/libfont/Makefile
```

```
 create mode 100644 navy-apps/libs/libos/Makefile
 create mode 100644 navy-apps/libs/libos/README.md
 create mode 100644 navy-apps/tests/bmp/Makefile
 create mode 100644 navy-apps/tests/dummy/Makefile
 create mode 100644 navy-apps/tests/events/Makefile
 create mode 100644 navy-apps/tests/hello/Makefile
 create mode 100644 navy-apps/tests/text/Makefile
 create mode 100644 navy-apps/tests/videotest/Makefile
zhaoweikang@zhaoweikang:~/ics2017$ sudo git log
commit 8a852335191cf80edb25996de234c03b50e51a72
Author: 161630220-Zhao Weikang <2875206963@qq.com>
Date:   Sun May 13 17:32:32 2018 +0800

    fix bug for pa3.1

commit 07761d61cc4754dbaaf44a8ad7c6a53d5676d833
Author: 161630220-Zhao Weikang <2875206963@qq.com>
Date:   Thu May 10 09:17:52 2018 +0800

    before starting pa3

commit ea21a2efaa92091bb5d631d93587313c95e8c450
Author: 161630220-Zhao Weikang <2875206963@qq.com>
Date:   Wed May 2 20:50:40 2018 +0800

    fix bug for pa2.3

commit cc11412df2700fdc325252df017d0fbed7965cbb
Author: 161630220-Zhao Weikang <2875206963@qq.com>
Date:   Fri Apr 6 16:48:19 2018 +0800

    before starting pa2
```