

南京航空航天大学《计算机组成原理II课程设计》报告

- 姓名：唐希
- 班级：1619201
- 学号：161920122
- 报告阶段：PA1.1
- 完成日期：2021.3.28
- 本次实验，我完成了大部分内容。

目录

南京航空航天大学《计算机组成原理II课程设计》报告

目录

思考题和git

- 1.有什么办法
- 2.一些简单的正则表达式
- 3.这是为什么？
- 4.如何处理以上问题？
- 5.递归求值的过程？
- 6.体验监视点
- 7.科学起名
- 8.温故而知新
- 9.一点也不能长
- 10.随心所欲的断点
- 11.NEMU 的前世今生
12. 尝试通过目录定位关注的问题
13. 理解基础设施
14. 查阅i386手册
15. shell 命令
16. 使用 `man`
17. `git log` 和远程git仓库提交截图

操作题

PA1.2 表达式求值

- 任务1：编写匹配规则(1)
- 任务2：添加p命令
- 任务3：识别并存储 token
- 任务4：实现括号匹配
- 任务5：寻找当前子表达式的中心操作符
- 任务6：编写匹配规则，实现表达式求值
 - 编写匹配规则（2）
 - 编写 `eval()` 函数
 - 测试表达式求值
- 任务7：实现指针解引用
- 任务8（选做）：实现负数

PA1.3监视点

- 任务1：监视点结构体
- 任务2：监视点池管理

任务3: 将监视点加入调试器功能

任务4: 实现监视点

任务5: 使用模拟断点

遇到的问题及解决办法

实验心得

其他备注

思考题和git

1.有什么办法

对表达式字符序列自左向右进行扫描，通过按顺序把操作数与操作符分别入栈。遇到操作数进入操作数栈。遇到运算符，首先与运算符栈的栈顶运算符比较优先级，若栈顶元素符的优先级高于当前运算符，则栈顶运算符出栈并执行运算，否则将当前运算符入运算符栈，直至整个表达式求值完毕。

2.一些简单的正则表达式

- 以 0x 开头的 32 位十六进制整数:

- `0x[0-9a-fA-F]{32}`

- 英文字母和数字组成的字符串:

- `[0-9a-zA-Z]+`

- C 语言中的变量名或函数名

- `^[a-zA-Z_][a-zA-Z0-9_]*$`

- 学号 - 姓名 - PA1.1.pdf

- `\d{9}[-][\u4e00-\u9fa5]{1,10}[-][PA1.1.pdf]`

3.这是为什么?

一个\代表转义，第一个\用来对后面的\转义。

4.如何处理以上问题?

```
int j;
for (j = 0; j < NR_REGEX; j++) {
    memset(tokens[j].str, '\0', 32);
    tokens[j].type = 0;
}
```

在每次调用make_tokens的开头将所有str的成员置0。

5.递归求值的过程?

```
<expr> ::= <expr> + <term>
        | <expr> - <term>
        | <term>

<term>  ::= <term> * <factor>
        | <term> / <factor>
        | <factor>

<factor> ::= ( <expr> )
         | Num
```

在四则运算中，双括号()的优先级最高，乘除运算*/的优先级其次，加减运算的优先级最低。一个表达式是先分解成±运算，然后分解成乘除运算*/，最后再分解为Expr，根据之前入栈操作数和操作符来完成对四则运算优先级的匹配。

6.体验监视点

tangxi@debian: ~/ics2021

```
Reading symbols from sum...done.
(gdb) l
1      #include<stdio.h>
2      int i,j,sum;
3      int main()
4      {
5          sum=0;
6          for(i=0;i<4;i++)
7              j=0;
8              sum=i+j;
9          return 0;
10     }
(gdb) watch i
Hardware watchpoint 1: i
(gdb) watch sum
Hardware watchpoint 2: sum
(gdb) run
Starting program: /home/tangxi/ics2021/sum

Hardware watchpoint 1: i

Old value = 0
New value = 1
0x004011cf in main () at sum.c:6
6          for(i=0;i<4;i++)
(gdb) d 0
warning: bad breakpoint number at or near '0'
(gdb) info break
Num      Type           Disp Enb Address      What
1        hw watchpoint  keep y                i
        breakpoint already hit 1 time
2        hw watchpoint  keep y                sum
(gdb) d 1
(gdb) info w
Ambiguous info command "w": warranty, watchpoints, win.
(gdb) info break
Num      Type           Disp Enb Address      What
2        hw watchpoint  keep y                sum
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/tangxi/ics2021/sum

Hardware watchpoint 2: sum

Old value = 0
New value = 4
main () at sum.c:9
9          return 0;
(gdb)
```

7.科学起名

数据的储存有大端与小端两种方式，小端字节序，低地址低字节，高地址高字节；大端字节序，低地址低字节，高地址低字节，所以单字节和4字节打印出来的顺序不一样。

8.温故而知新

static是静态全局变量，该变量只能被本文件中的函数调用，并且是全局变量，而不能被同一程序其他文件中的函数调用。在此处使用static是为了避免它被误修改。

9.一点也不能长

这是必须的。因为在gdb中对代码的某一行设置断点时，会先保存第一个字节，然后写入一条INT 3指令，机器码为0xcc，仅有一个字节，设置和取消断点时也需要保存和恢复一个字节。

文章中断点机制不能正常工作。因为INT3断点将被调试进程中对地址处的字节替换为0xcc，当int3指令的长度变成2个字节，其他指令同x86，会导致这里本来指令的第一个字节被2个字节所代替，空间不够，不正确。

10.随心所欲的断点

将断点设置在指令的非首字节，就无法检测到断点。因为会检测函数的地址，读取它的第一个字节，判断是否等于0xCCH。

11.NEMU 的前世今生

调试器是在真机上运行的调试工具，用来设置断点，查看变量的数值。因为在具体的机器上进行调试操作，所以能更加直接，快速，清晰地反映出各种问题。而且模拟器调试在支持方面也比调试器弱。

12. 尝试通过目录定位关注的问题

CHAPTER 5 MEMORY MANAGEMENT	91
5.1 SEGMENT TRANSLATION	92
5.1.1 Descriptors.....	92
5.1.2 Descriptor Tables.....	94
5.1.3 Selectors	96
5.1.4 Segment Registers	97
5.2 PAGE TRANSLATION	98
5.2.1 Page Frame.....	98
5.2.2 Linear Address.....	98

看目录可以找到 selector 的位置

13. 理解基础设施

1: $500 \times 45 \times 20 \times 0.5 / 60 = 75$ ，要花费75个小时

2: $75 \times 3 / 2 = 50$,节省了50个小时

14. 查阅i386手册

1、EFLAGS寄存器中的CF位是什么意思？

P34查看附录C，CF是进位标志；
P419附录C

2.3.4.1 Status Flags

The status flags of the EFLAGS register allow the results of one instruction to influence later instructions. The arithmetic instructions use **OF, SF, ZF, AF, PF, and CF**. The SCAS (Scan String), CMPS (Compare String), and LOOP instructions use ZF to signal that their operations are complete. There are instructions to set, clear, and complement CF before execution of an arithmetic instruction. Refer to Appendix C for definition of each status flag.

Status Flags' Functions

Bit	Name	Function
0	CF	Carry Flag — Set on high-order bit carry or borrow; cleared otherwise.
2	PF	Parity Flag — Set if low-order eight bits of result contain an even number of 1 bits; cleared otherwise.
4	AF	Adjust flag — Set on carry from or borrow to the low order four bits of AL; cleared otherwise. Used for decimal arithmetic.
6	ZF	Zero Flag — Set if result is zero; cleared otherwise.
7	SF	Sign Flag — Set equal to high-order bit of result (0 is positive, 1 if negative).
11	OF	Overflow Flag — Set if result is too large a positive number or too small a negative number (excluding sign-bit) to fit in destination operand; cleared otherwise.

2、ModR/M字节是什么？

P241,17.2.1

ModR/M由Mod, Reg/Opcode, R/M三部分组成。Mod字段占据数据的两个最高有效位字节12位，与r/m字段组合形成32个可能值，提供寄存器寻址和内存寻址，reg字段，占据mod后面的三位字段，定操作码的寄存器号或更多三位信息。reg字段的含义由第一个（操作码）指令的字节。r/m字段，它占据数据的三个最低有效位字节，可以指定寄存器作为操作数的位置，也可以形成寻址方式编码的一部分，与字段相结合。

3、mov指令的具体格式是怎么样的？

P347格式是DEST ←SRC


15. shell 命令

在nemu下输入 `find . -name "*[.h|.cpp]" | xargs wc -l`

 tangxi@debian: ~/ics2021/nemu

```
56 ./include/cpu/reg.h
7 ./include/monitor/monitor.h
23 ./include/monitor/watchpoint.h
8 ./include/monitor/expr.h
wc: ./build/obj/misc: Is a directory
0 ./build/obj/misc
wc: ./build/obj/cpu/exec: Is a directory
0 ./build/obj/cpu/exec
wc: ./src: Is a directory
0 ./src
12 ./src/main.c
28 ./src/device/timer.c
38 ./src/device/vga.c
98 ./src/device/device.c
29 ./src/device/serial.c
70 ./src/device/keyboard.c
56 ./src/device/io/port-io.c
70 ./src/device/io/mmio.c
wc: ./src/misc: Is a directory
0 ./src/misc
31 ./src/misc/logo.c
28 ./src/memory/memory.c
13 ./src/cpu/intr.c
64 ./src/cpu/reg.c
113 ./src/cpu/decode/modrm.c
311 ./src/cpu/decode/decode.c
wc: ./src/cpu/exec: Is a directory
0 ./src/cpu/exec
66 ./src/cpu/exec/logic.c
65 ./src/cpu/exec/system.c
46 ./src/cpu/exec/special.c
32 ./src/cpu/exec/cc.c
223 ./src/cpu/exec/arith.c
8 ./src/cpu/exec/all-instr.h
43 ./src/cpu/exec/control.c
259 ./src/cpu/exec/exec.c
9 ./src/cpu/exec/prefix.c
77 ./src/cpu/exec/data-mov.c
44 ./src/monitor/cpu-exec.c
359 ./src/monitor/debug/expr.c
244 ./src/monitor/debug/ui.c
169 ./src/monitor/debug/watchpoint.c
314 ./src/monitor/diff-test/protocol.c
157 ./src/monitor/diff-test/diff-test.c
48 ./src/monitor/diff-test/protocol.h
106 ./src/monitor/diff-test/gdb-host.c
143 ./src/monitor/monitor.c
39 ./runall.sh
4085 total
tangxi@debian:~/ics2021/nemu$
```

输入git checkout master, 回到master分支

 tangxi@debian: ~/ics2021/nemu

```
60 ./include/cpu/reg.h
7 ./include/monitor/monitor.h
15 ./include/monitor/watchpoint.h
8 ./include/monitor/expr.h
wc: ./build/obj/misc: Is a directory
0 ./build/obj/misc
wc: ./build/obj/cpu/exec: Is a directory
0 ./build/obj/cpu/exec
wc: ./src: Is a directory
0 ./src
12 ./src/main.c
28 ./src/device/timer.c
38 ./src/device/vga.c
98 ./src/device/device.c
29 ./src/device/serial.c
70 ./src/device/keyboard.c
56 ./src/device/io/port-io.c
70 ./src/device/io/mmio.c
wc: ./src/misc: Is a directory
0 ./src/misc
31 ./src/misc/logo.c
28 ./src/memory/memory.c
13 ./src/cpu/intr.c
43 ./src/cpu/reg.c
113 ./src/cpu/decode/modrm.c
311 ./src/cpu/decode/decode.c
wc: ./src/cpu/exec: Is a directory
0 ./src/cpu/exec
66 ./src/cpu/exec/logic.c
65 ./src/cpu/exec/system.c
46 ./src/cpu/exec/special.c
32 ./src/cpu/exec/cc.c
223 ./src/cpu/exec/arith.c
8 ./src/cpu/exec/all-instr.h
43 ./src/cpu/exec/control.c
259 ./src/cpu/exec/exec.c
9 ./src/cpu/exec/prefix.c
77 ./src/cpu/exec/data-mov.c
44 ./src/monitor/cpu-exec.c
109 ./src/monitor/debug/expr.c
116 ./src/monitor/debug/ui.c
23 ./src/monitor/debug/watchpoint.c
314 ./src/monitor/diff-test/protocol.c
157 ./src/monitor/diff-test/diff-test.c
48 ./src/monitor/diff-test/protocol.h
106 ./src/monitor/diff-test/gdb-host.c
143 ./src/monitor/monitor.c
39 ./runall.sh
3536 total
tangxi@debian:~/ics2021/nemu$
```


通过相减可以知道写了多少代码

再回到pa1分支

输入 `find . -name "*.cpp|.h" | xargs grep "^." | wc -l`

```
tangxi@debian: ~/ics2021/nemu
tangxi@debian:~/ics2021/nemu$ cd ..
tangxi@debian:~/ics2021$ cd nemu/
tangxi@debian:~/ics2021/nemu$
tangxi@debian:~/ics2021/nemu$ * pa1
-bash: build: command not found
tangxi@debian:~/ics2021/nemu$ tangxi@debian:~/ics2021$ cd nemu/
-bash: tangxi@debian:~/ics2021$: No such file or directory
tangxi@debian:~/ics2021/nemu$ find . -name "*.cpp|.h" | xargs grep "^." | wc -l
-bash: .h": command not found
0
tangxi@debian:~/ics2021/nemu$ find . -name "*.cpp|.h" | xargs grep "^." | wc -l
-bash: .h": command not found
0
tangxi@debian:~/ics2021/nemu$ find . -name "*.cpp|.h" | xargs grep "^." | wc -l
-bash: .h": command not found
0
tangxi@debian:~/ics2021/nemu$ find . -name "*.cpp|.h" | xargs grep "^." | wc -l
find: '-name': No such file or directory
find: '*.cpp|.h': No such file or directory
grep: .: Is a directory
grep: ./include: Is a directory
grep: ./include/device: Is a directory
grep: ./include/memory: Is a directory
grep: ./include/cpu: Is a directory
grep: ./include/monitor: Is a directory
grep: ./build: Is a directory
grep: ./build/obj: Is a directory
grep: ./build/obj/device: Is a directory
grep: ./build/obj/device/io: Is a directory
grep: ./build/obj/misc: Is a directory
grep: ./build/obj/memory: Is a directory
grep: ./build/obj/cpu: Is a directory
grep: ./build/obj/cpu/decode: Is a directory
grep: ./build/obj/cpu/exec: Is a directory
grep: ./build/obj/monitor: Is a directory
grep: ./build/obj/monitor/debug: Is a directory
grep: ./build/obj/monitor/diff-test: Is a directory
grep: ./src: Is a directory
grep: ./src/device: Is a directory
grep: ./src/device/io: Is a directory
grep: ./src/misc: Is a directory
grep: ./src/memory: Is a directory
grep: ./src/cpu: Is a directory
grep: ./src/cpu/decode: Is a directory
grep: ./src/cpu/exec: Is a directory
grep: ./src/monitor: Is a directory
grep: ./src/monitor/debug: Is a directory
grep: ./src/monitor/diff-test: Is a directory
3533
tangxi@debian:~/ics2021/nemu$
```

切到master, 同理操作

 tangxi@debian: ~/ics2021

```
grep: ../git/objects/58: Is a directory
grep: ../git/objects/5a: Is a directory
grep: ../git/objects/11: Is a directory
grep: ../git/objects/fd: Is a directory
grep: ../git/objects/6e: Is a directory
grep: ../git/objects/ed: Is a directory
grep: ../git/objects/96: Is a directory
grep: ../git/objects/7d: Is a directory
grep: ../git/objects/27: Is a directory
grep: ../git/objects/ce: Is a directory
grep: ../git/objects/89: Is a directory
grep: ../git/objects/29: Is a directory
grep: ../git/objects/a9: Is a directory
grep: ../git/objects/cd: Is a directory
grep: ../git/objects/b7: Is a directory
grep: ../git/objects/c7: Is a directory
grep: ../git/objects/68: Is a directory
grep: ../git/objects/40: Is a directory
grep: ../git/objects/1f: Is a directory
grep: ../git/objects/ca: Is a directory
grep: ../git/objects/c8: Is a directory
grep: ../git/objects/98: Is a directory
grep: ../git/objects/28: Is a directory
grep: ../git/objects/51: Is a directory
grep: ../git/objects/50: Is a directory
grep: ../git/objects/5c: Is a directory
grep: ../git/objects/31: Is a directory
grep: ../git/objects/f7: Is a directory
grep: ../git/objects/5f: Is a directory
grep: ../git/objects/a1: Is a directory
grep: ../git/objects/c3: Is a directory
grep: ../git/objects/e7: Is a directory
grep: ../git/objects/33: Is a directory
grep: ../git/objects/86: Is a directory
grep: ../git/objects/10: Is a directory
grep: ../git/objects/48: Is a directory
grep: ../git/objects/56: Is a directory
grep: ../git/objects/1a: Is a directory
grep: ../git/objects/d1: Is a directory
grep: ../git/objects/7e: Is a directory
grep: ../git/objects/54: Is a directory
grep: ../git/objects/b6: Is a directory
grep: ../git/objects/6c: Is a directory
grep: ../git/objects/23: Is a directory
grep: ../git/objects/6b: Is a directory
grep: ../git/objects/da: Is a directory
grep: ../git/objects/0d: Is a directory
grep: ../git/objects/5d: Is a directory
```

207758


16. 使用 man

Wall 使GCC产生尽可能多的警告信息，取消编译操作，打印出编译时所有错误或警告信息。

Werror 要求GCC将所有的警告当成错误进行处理，取消编译操作。

使用-Wall和-Werror就是为了找出存在的错误，尽可能地避免程序运行出错。

17. git log 和远程git仓库提交截图

 tangxi@debian: ~/ics2021

```
commit 550395198d8aa11c04d07d1ab888f0b4abaa0918 (HEAD -> pal)
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Apr 19 11:17:52 2021 +0800

    > run
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
    11:17:52 up 1:21, 1 user, load average: 0.00, 0.00, 0.00
    3192e76dc9a0c2e1863675ec2288f0b57a1e437f

commit 6120e8e382cdb1531ddba7af7a7bee721e59e281
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Apr 19 11:17:02 2021 +0800

    > run
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
    11:17:02 up 1:20, 1 user, load average: 0.00, 0.00, 0.00
    df13bcef28d8ac29b78c45adfd3bb65217a9476b


commit 9b1d49557151284b4e972688c5141c908f1d82dc
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Apr 19 11:13:15 2021 +0800

    > run
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
    11:13:15 up 1:16, 1 user, load average: 0.08, 0.02, 0.01
    f2eb26f0a2d27be423a1e631ff666d6d177a8d69

commit f5f7bb767b58e8e29ce998189dda39bb43883701
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Apr 19 11:13:15 2021 +0800

    > compile
    161920122
    tangxi
    Linux debian 4.19.0-14-686 #1 SMP Debian 4.19.171-2 (2021-01-30) i686 GNU/Linux
    11:13:15 up 1:16, 1 user, load average: 0.08, 0.02, 0.01
    4a54731b19b5be7c755110184c9050a3dd4dec

commit cbb943ad8fe31668ea5e3b2ad2b7fb08b6a3a040
Author: tracer-ics2017 <tracer@njuics.org>
Date:   Mon Apr 19 10:58:21 2021 +0800

    > run
: 
```

```

tangxi@debian: ~/ics2021
make[2]: Entering directory '/home/tangxi/ics2021/navy-apps/tests/text'
rm -rf /home/tangxi/ics2021/navy-apps/tests/text/build/
make[2]: Leaving directory '/home/tangxi/ics2021/navy-apps/tests/text'
make[2]: Entering directory '/home/tangxi/ics2021/navy-apps/tests/videotest'
rm -rf /home/tangxi/ics2021/navy-apps/tests/videotest/build/
make[2]: Leaving directory '/home/tangxi/ics2021/navy-apps/tests/videotest'
rm -f fsimg/bin/*
make[1]: Leaving directory '/home/tangxi/ics2021/navy-apps'
git gc
Enumerating objects: 1414, done.
Counting objects: 100% (1414/1414), done.
Compressing objects: 100% (1152/1152), done.
Writing objects: 100% (1414/1414), done.
Total 1414 (delta 426), reused 1133 (delta 216)
cd .. && tar cj ics2021 > 161920122.tar.bz2
tangxi@debian:~/ics2021$ git push
fatal: The current branch pal has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin pal

tangxi@debian:~/ics2021$ git push myrepo pal
fatal: unable to access 'https://e.coding.net/tangxil/tangxi/PA.git/': Could not resolve host: e.coding.net
tangxi@debian:~/ics2021$ ^C
tangxi@debian:~/ics2021$ git push myrepo pal
fatal: unable to access 'https://e.coding.net/tangxil/tangxi/PA.git/': Could not resolve host: e.coding.net
tangxi@debian:~/ics2021$ git push myrepo pal
Username for 'https://e.coding.net': 231816
Password for 'https://231816@e.coding.net':
remote: CODING 提示: Authentication failed.
remote: 认证失败, 请确认您输入了正确的账号密码。
fatal: Authentication failed for 'https://e.coding.net/tangxil/tangxi/PA.git/'
tangxi@debian:~/ics2021$ git push myrepo pal
Username for 'https://e.coding.net': tangxi
Password for 'https://tangxi@e.coding.net':
remote: CODING 提示: Authentication failed.
remote: 认证失败, 请确认您输入了正确的账号密码。
fatal: Authentication failed for 'https://e.coding.net/tangxil/tangxi/PA.git/'
tangxi@debian:~/ics2021$ git push myrepo pal
Username for 'https://e.coding.net': 15913121302
Password for 'https://15913121302@e.coding.net':
Enumerating objects: 192, done.
Counting objects: 100% (192/192), done.
Compressing objects: 100% (52/52), done.
Writing objects: 100% (175/175), 22.48 KiB | 2.50 MiB/s, done.
Total 175 (delta 133), reused 162 (delta 121)
remote: Resolving deltas: 100% (133/133), completed with 10 local objects.
To https://e.coding.net/tangxil/tangxi/PA.git
 * [new branch]      pal -> pal
tangxi@debian:~/ics2021$ s

```

tracer-ics2017 > compile			最后提交 7c549cc795 于 1 小时前
..			
expr.c	tracer-ics2...	> compile	16 小时前
ui.c	tracer-ics2...	> compile	1 小时前
watchpoint.c	tracer-ics2...	> compile	1 小时前

操作题

PA1.2 表达式求值

任务1: 编写匹配规则(1)

编写要求的正则表达式, 添加运算符

```

enum {
    TK_NOTYPE = 256, TK_EQ,
    TK_PLUS,

    /* TODO: Add more token types */

    TK_DEC, TK_HEX,
    TK_REG,
    TK_NEQ,
    TK_AND,

```

```

TK_OR,
TK_NOT,
TK_DEREF,
TK_NEG

};

static struct rule {
    char *regex;
    int token_type;
} rules[] = {

    /* TODO: Add more rules.
     * Pay attention to the precedence level of different rules.
     */

    {" +", TK_NOTYPE},      // spaces
    {"\\+", '+'},          // plus
    {"==", TK_EQ},         // equal
    {"\\-", '-'},          // minus
    {"\\*", '*'},          // multiply
    {"\\/", '/'},          // divide
    {"\\$[a-z]{2,3}", TK_REG}, // regex for x86 register

    {"\\b[0-9]+\\b", TK_DEC}, // decimal-number
    {"0[xX][0-9a-fA-F]+", TK_HEX}, // hexadecimal-number

    {"-", '-'},            // sub
    {"\\*", '*'},          // mul
    {"\\/", '/'},          // div
    {"\\(", '('},          //
    {"\\)", ')'},          //
    {"!=", TK_NEQ},        // not equal
    {"&&", TK_AND},         // and
    {"\\|\\|", TK_OR},      // or
    {"!", TK_NOT},         // not

};

```

任务2：添加p命令

进入nemu/src/monitor/debug/ui.c中，添加p命令。

char*类型的args代表求表达式，当他不为空时调用位于expr.c的expr函数求出结果，输出16、10进制结果。

源代码：

```

static int cmd_p(char *args) {
    bool success;
    if(args==NULL)
        printf("No Result!");
    else{
        success=true;
        uint32_t result = expr(args,&success);
        printf("result = 0x%x %d\n", result, result);
    }
    return 0;
}

```

运行情况

```

./build/nemu -i ./build/nemu-log.txt
[src/monitor/monitor.c,47,load_default_img] No image is given. Use the default build-in image.
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 07:49:56, Apr 15 2021
For help, type "help"
(nemu) p 1+1
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 0 with len 1: 1
[src/monitor/debug/expr.c,114,make_token] match rules[1] = "+" at position 1 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 2 with len 1: 1
result = 0x2 2
(nemu)

```

任务3：识别并存储 token

再switch语句中添加各个操作符，并且判断十进制和十六进制操作数输入时是否超过了str的最大位数32位。当表达式中的字符合法时，由switch选择结构进入下一步，否则break。

```

static bool make_token(char *e) {
    int position = 0;
    int i;
    regmatch_t pmatch;

    nr_token = 0;

    int j;
    for (j = 0; j < NR_REGEX; j++) {
        memset(tokens[j].str, '\0', 32);
        tokens[j].type = 0;
    }

    while (e[position] != '\0') {
        /* Try all rules one by one. */
        for (i = 0; i < NR_REGEX; i++) {
            if (regexexec(&re[i], e + position, 1, &pmatch, 0) == 0 && pmatch.rm_so == 0) {
                char *substr_start = e + position;
                int substr_len = pmatch.rm_eo;

                Log("match rules[%d] = \"%s\" at position %d with len %d: %.*s",
                    i, rules[i].regex, position, substr_len, substr_len, substr_start);
                position += substr_len;
                if (rules[i].token_type == TK_NOTYPE) break;
                switch (rules[i].token_type) {
                    case TK_DEC:
                    case TK_HEX:
                        if(substr_len > 32) {

```



```

        printf("ERROR! the length of num in this
expr is longer than 32bit\n");

        assert(0);

    }

    case '+':
    case '-':
    case '*':
    case '/':
    case TK_AND:
    case TK_OR:
    case TK_NOT:
    case TK_REG:
    case '(':
    case ')':
    case TK_EQ:
    case TK_REG:
    case TK_NEQ:
//      case TK_POINTER: /* TODO: implement it*/
        strncpy(tokens[nr_token].str, substr_start,
substr_len);

        tokens[nr_token].type = rules[i].token_type;
        nr_token++;
        break;

    default: TODO();
}

break;
}
}

if (i == NR_REGEX) {
    printf("no match at position %d\n%s\n%*.s^\n", position, e, position, "");
    return false;
}
}

return true;
}

```

任务4：实现括号匹配

从给定的开始位置p开始顺序检查str成员，如果开始不是左括号或者右括号的话，返回false。当还有没检查的元素时，判断tokens[p].str是不是左括号（“或者右括号”），当匹配到左括号时，标志数flag加1，匹配到右括号时flag减1。检查到最后若flag不为0，则返回false，否则返回true。

源代码：


```

bool check_parentheses(int p ,int q){
    int i,tag = 0;
    if(tokens[p].type != TK_LEFT || tokens[q].type != TK_RIGHT) return false; //
    首尾没有()则为false
    for(i = p ; i <= q ; i ++){
        if(strcmp(tokens[p].str,"(") == 0) tag++;
        else if(strcmp(tokens[p].str,")") == 0) tag--;
        if(tag == 0 && i < q) return false ; //(4 + 3) * (2 - 1) 返回false
    }
    if( tag != 0 ) return false;
    return true;
}

```

任务5：寻找当前子表达式的中心操作符

给定开始位置p和结束位置q，由p到q，首先检查表达式中的大小括号是否匹配。如果匹配的话就判断是不是四组运算、指针应用和其他运算符。当排除掉括号优先级的匹配后，只需要匹配第一个字符就时最后进行操作的中心操作符。

```

uint32_t find_dominated_op(uint32_t p, uint32_t q) {
    assert(p <= q);
    uint32_t i = p;
    uint32_t op = p;
    int pnum = 0;

    for (i = p; i < q; i++)
    {

        if(strcmp(tokens[i].str, "(") == 0) pnum++;
        if(strcmp(tokens[i].str, ")") == 0) pnum--;

        if (pnum == 0)
            if (strcmp(tokens[i].str, "+") == 0 || strcmp(tokens[i].str, "-") ==
0 ||
                strcmp(tokens[i].str, "*") == 0 || strcmp(tokens[i].str, "/") ==
0 ||
                tokens[i].type == TK_DEREF || /* pointer */
                tokens[i].type == TK_EQ || tokens[i].type == TK_NEQ ||
tokens[i].type == TK_AND ||
                tokens[i].type == TK_OR || tokens[i].type == TK_NOT)
            {
                op = i;
                break;
            }
    }
    return op;
}

```

①注意：出现形式如 $3*1+(2+3)$ 这种非法括号时可能会出错。

任务6：编写匹配规则，实现表达式求值

编写匹配规则 (2)

见上文任务1；

编写 eval() 函数

先检查输入的p和q是否合法，然后将tokens转化位无符号长整型数，p==q,即只有一个token的时候返回对应的uint_t类型的变量。如果能完成括号匹配，那么往括号里面继续递归，否则用寻找中心操作符算法给出计算的逆序，知道遍历完整个表达式，最后进行计算。

```
uint32_t eval(uint32_t p, uint32_t q) {

    if (p > q) {
        printf("Worng expression!");
        assert(0);
    }

    if (p == q) {
        uint32_t value = 0;
        if (tokens[p].type == TK_DEC)
            value = strtoul(tokens[p].str, NULL, 10);

        if (tokens[p].type == TK_HEX)
            value = strtoul(tokens[p].str, NULL, 16);

        if (tokens[p].type == TK_REG) {
            int i;
            for (i = 0; i < strlen(tokens[p].str); i++) tokens[p].str[i] =
tokens[p].str[i+1];
            bool success = true;
            value = isa_reg_str2val(tokens[p].str, &success);
        }
        /*in reg.c
        uint32_t isa_reg_str2val (const char * s, bool * success) {
            return 0 ;
        }*/

        //if (tokens[p].type == TK_DEREF)

        return value;

    } else if (check_parentheses(p, q) == true) {
        // printf("check parentheses is true, p = %u, q = %u\n", p, q);
        return eval(p + 1, q - 1);

    } else {

        int op = getOpPosition(p, q); /* get the position of the op in the
subexpression */

        // printf("get op position = %u, op = %s\n",op, tokens[op].str);

        uint32_t val1 = 0;
        if(tokens[op].type != TK_DEREF &&
            tokens[op].type != TK_NEG &&
```

```

        tokens[op].type != TK_NOT)
        val1 = eval(p, op - 1);

        uint32_t val2 = eval(op + 1, q);

        // printf("[eval debug]op = %s, val1 = %u, val2 = %u\n", tokens[op].str,
        val1, val2);

        switch (tokens[op].type) {
            case '+': return val1 + val2;
            case '-': return val1 - val2;
            case '*': return val1 * val2;
            case '/': return val1 / val2; /* TODO: what about val2 is zero? */
            case TK_DEREF: return vaddr_read(val2,4);
            case TK_EQ: return val1 == val2 ? 1 : 0;
            case TK_NEQ: return val1 != val2 ? 1 : 0;
            case TK_AND: return val1 && val2 ? 1 : 0;
            case TK_OR: return val1 || val2 ? 1 : 0;
            case TK_NOT: return !val2;
            case TK_NEG: return -val2;
            default: assert(0);
        }

    }

}

return 0;
}

bool certain_type_for_not(int type) {
    switch(type) {
        case '+':
        case '-':
        case '*':
        case '/':
        case '(': return true;
        default : return false;
    }
}
}

```

测试表达式求值

测试样例：

```

Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 07:49:56, Apr 15 2021
For help, type "help"
(nemu) p $eax
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\\$[a-z]{2,3}" at position 0 with len 4: $eax
result = 0x506c4148 1349271880
(nemu) p $eip
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\\$[a-z]{2,3}" at position 0 with len 4: $eip
result = 0x100000 1048576
(nemu) p *0x100000
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "\\*" at position 0 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[8] = "0[xX][0-9a-fA-F]+" at position 1 with len 8: 0x100000
result = 0x1234b8 1193144
(nemu) *$eip
Unknown command '*$eip'
(nemu) p *$eip
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "\\*" at position 0 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\\$[a-z]{2,3}" at position 1 with len 4: $eip
result = 0x1234b8 1193144
(nemu) p 2 * ($eax + $ebx)
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 0 with len 1: 2
[src/monitor/debug/expr.c,114,make_token] match rules[0] = "+" at position 1 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "\\*" at position 2 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[0] = "+" at position 3 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[12] = "\\(" at position 4 with len 1: (
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\\$[a-z]{2,3}" at position 5 with len 4: $eax
[src/monitor/debug/expr.c,114,make_token] match rules[0] = "+" at position 9 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[1] = "\\+" at position 10 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[0] = "+" at position 11 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\\$[a-z]{2,3}" at position 12 with len 4: $ebx
[src/monitor/debug/expr.c,114,make_token] match rules[13] = "\\)" at position 16 with len 1: )
result = 0x6e0b7ac4 1846246084
(nemu) 

```

自己的测试样例:

```

./build/nemu -f ./build/nemu-log.txt
[src/monitor/monitor.c,47,load_default_img] No image is given. Use the default build-in image.
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 07:49:56, Apr 15 2021
For help, type "help"
(nemu) p 1+1
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 0 with len 1: 1
[src/monitor/debug/expr.c,114,make_token] match rules[1] = "\\+" at position 1 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 2 with len 1: 1
result = 0x2 2
(nemu) 

```

```

(nemu) p 1+1
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 0 with len 1: 1
[src/monitor/debug/expr.c,114,make_token] match rules[1] = "\\+" at position 1 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 2 with len 1: 1
result = 0x2 2
(nemu) p 2+2*3
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 0 with len 1: 2
[src/monitor/debug/expr.c,114,make_token] match rules[1] = "\\+" at position 1 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 2 with len 1: 2
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "\\*" at position 3 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 4 with len 1: 3
result = 0x8 8
(nemu) p (2+2)*3
[src/monitor/debug/expr.c,114,make_token] match rules[12] = "\\(" at position 0 with len 1: (
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 1 with len 1: 2
[src/monitor/debug/expr.c,114,make_token] match rules[1] = "\\+" at position 2 with len 1: +
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 3 with len 1: 2
[src/monitor/debug/expr.c,114,make_token] match rules[13] = "\\)" at position 4 with len 1: )
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "\\*" at position 5 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 6 with len 1: 3
result = 0xc 12
(nemu) p (5-3)/2
[src/monitor/debug/expr.c,114,make_token] match rules[12] = "\\(" at position 0 with len 1: (
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 1 with len 1: 5
[src/monitor/debug/expr.c,114,make_token] match rules[3] = "\\-" at position 2 with len 1: -
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 3 with len 1: 3
[src/monitor/debug/expr.c,114,make_token] match rules[13] = "\\)" at position 4 with len 1: )
[src/monitor/debug/expr.c,114,make_token] match rules[5] = "\\/" at position 5 with len 1: /
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\\b[0-9]+\\b" at position 6 with len 1: 2
result = 0x1 1
(nemu) 

```

```
(nemu) p 3!=3
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 0 with len 1: 3
[src/monitor/debug/expr.c,114,make_token] match rules[14] = "!=" at position 1 with len 2: !=
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 3 with len 1: 3
result = 0x0 0
(nemu) p 3==3
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 0 with len 1: 3
[src/monitor/debug/expr.c,114,make_token] match rules[2] = "==" at position 1 with len 2: ==
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 3 with len 1: 3
result = 0x1 1
(nemu) []
```

```
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 07:49:56, Apr 15 2021
For help, type "help"
(nemu) p !1
[src/monitor/debug/expr.c,114,make_token] match rules[17] = "!" at position 0 with len 1: !
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 1 with len 1: 1
result = 0x0 0
(nemu) p !0
[src/monitor/debug/expr.c,114,make_token] match rules[17] = "!" at position 0 with len 1: !
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 1 with len 1: 0
result = 0x1 1
(nemu) p 1&&0
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 0 with len 1: 1
[src/monitor/debug/expr.c,114,make_token] match rules[15] = "&&" at position 1 with len 2: &&
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 3 with len 1: 0
result = 0x0 0
(nemu) p 1&&1
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 0 with len 1: 1
[src/monitor/debug/expr.c,114,make_token] match rules[15] = "&&" at position 1 with len 2: &&
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 3 with len 1: 1
result = 0x1 1
(nemu) p 1||0
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 0 with len 1: 1
[src/monitor/debug/expr.c,114,make_token] match rules[16] = "||" at position 1 with len 2: ||
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 3 with len 1: 0
result = 0x1 1
(nemu) p 0||0
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 0 with len 1: 0
[src/monitor/debug/expr.c,114,make_token] match rules[16] = "||" at position 1 with len 2: ||
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\b[0-9]+\b" at position 3 with len 1: 0
result = 0x0 0
(nemu) []
```

测试结果正确。

任务7：实现指针解引用

如果"*"是读入的第一个元素，或者在前一个token是一个操作数，那么就认为这个"*"是一个指针引用。在eval函数之前进行判断。

eval函数内：

```
switch (tokens[op].type) {
    +-----
    case TK_DEREF: return vaddr_read(va12,4);
    +-----
    default: assert(0);
}
```

```
bool certain_type_of_deref(int type) {
    switch(type) {
        case '+':
        case '-':
        case '*':
        case '/': return true;
        default : return false;
    }
}
```

expr函数内:

```
for (i = 0; i < nr_token; i++) {
    if (tokens[i].type == '*' && (i == 0 || certain_type_for_deref(tokens[i-1].type))) {
        tokens[i].type = TK_DEREF; //同样适用于其他单目运算符
    }
}
```

测试样例:

```
(nemu) p *$eip
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "\"*" at position 0 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\"$[a-z]{2,3}\" at position 1 with len 4: $eip
result = 0x1234b8 1193144
(nemu) []
```

任务8 (选做) : 实现负数

和指针解引用类似, 如果'-'是读入的第一个元素, 或者在前一个token是一个操作数, 那么就认为这个'-'是一个符号。然后对后面的一个操作数做取负指令neg。

expr函数内:

```
switch (tokens[op].type) {
    +-----
    case TK_NEG:    return -val2;
    +-----
    default: assert(0);
}
```

```
if (tokens[i].type == '-' && (i == 0 || certain_type_of_neg(tokens[i-1].type)))
{
    tokens[i].type = TK_NEG;
}
}
```

```
bool certain_type_of_neg(int type) { //负号匹配
    switch(type) {
        case '+':
        case '-':
        case '*':
        case '/':
        case '(': return true;
        default : return false;
    }
}
```

```
(nemu) p -(2*5)
[src/monitor/debug/expr.c,114,make_token] match rules[3] = "\"-" at position 0 with len 1: -
[src/monitor/debug/expr.c,114,make_token] match rules[12] = "\"(\" at position 1 with len 1: (
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\"b[0-9]+b\" at position 2 with len 1: 2
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "\"*" at position 3 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[7] = "\"b[0-9]+b\" at position 4 with len 1: 5
[src/monitor/debug/expr.c,114,make_token] match rules[13] = "\"\" at position 5 with len 1: )
result = 0xffffffff6 -10
(nemu) []
```

PA1.3监视点

任务1：监视点结构体

`EXPR` 储存要监视的表达式

`old_value` 表达式的旧值

`New_value` 表达式的新值，与旧值不一样的时候中断程序

```
typedef struct watchpoint {
    int NO;
    struct watchpoint *next;

    /* TODO: Add more members if necessary */

    // struct watchpoint *prev; /* the previous watchpoint */

    char EXPR[32];

    int old_value, New_value;
} WP;

bool watchpoint_monitor();
#endif
```

任务2：监视点池管理

`new_wp`是从`free`链表中取一个结点给`head`链表，且将表达式、值赋给它，修改开关，并输出该节点的编号。运用正则表达式判断是否为断点，若是则`type`为`b`，否则为`w`。

```
WP* new_wp(char *EXPR) {

    if (free_ == NULL) { /* there is no free wp */
        printf("there is no free wp!\n");
        return NULL;
    }

    f    bool success = true;
    expr(EXPR, &success);
    if (success == false) {
        printf("No symbol \"%s\" in current context. \n", EXPR);
        return NULL;
    }

    WP* node = free_;
    free_ = free_->next;

    /*insert the EXPR to it*/
    strcpy(node->EXPR, EXPR);

    node->next = head;
    head = node;
    return node;
}
```

free_wp函数是遍历head链表，先判断要归还的监视点是否超出了监视点池，如果超出了，则直接停止程序，未超出的话，则将该节点添加到free链表中。

```
bool free_wp(int N) {

    /* find the pointer to N */
    WP *wp = find_wp(N);

    WP *pre_wp = find_pre_wp(wp);

    if (wp == NULL) {
        printf("There is no %d watchpoint. \n", N);
        return false;
    }

    /* delete the wp from the list */
    if (wp == head) {
        head = NULL;
    } else {
        pre_wp->next = wp->next;
    }

    /* insert to free_ */
    wp->next = free_;
    /* move the free_ to wp */
    free_ = wp;

    printf("NO %d watchpoint has been deleted\n", N);
    return true; /* delete successfully */
}
```

任务3：将监视点加入调试器功能

在ics2019/nemu/src/monitor/debug/watchpoint.c和

ics2019/nemu/src/monitor/debug/ui.c中设置新命令

- **w** 命令：根据给予的表达式 `expr` 设置一个新的监视点，如 `w $eax`；

当输入w命令时，调用set_watchpoint函数，如果调用成功则返回创建的监视点编号。否则则输出提示信息。而info w则是调用new_wp函数，然后输出监视点的编号。


```
static int cmd_w(char *args) {

    char *EXPR = args;
    WP *wp = new_wp(EXPR);
    if(wp != NULL)
        printf("watchpoint %d: %s\n", wp->NO, wp->EXPR);
    else
        printf("Cannot creat watchpoint: %s\n", EXPR);

    return 0;

}
```

测试样例

```
+ LD build/nemu
./build/nemu -l ./build/nemu-log.txt
[src/monitor/monitor.c,47,load_default_img] No image is given. Use the default build-in image.
Welcome to NEMU!
[src/monitor/monitor.c,30,welcome] Build time: 11:13:14, Apr 19 2021
For help, type "help"
(nemu) w $eax
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\${a-z}{2,3}" at position 0 with len 4: $eax
watchpoint 0: $eax
```

- **d 命令**：根据给予的监视点编号 **NO** 删除该监视点，如 **d 1**；

当输入d命令时，用sscanf函数转化后调用free_wp函数，删除对应节点的监视点。

```
static int cmd_d(char *args) {

    char *argN = strtok(NULL, " "); // get the 'N'
    if(argN == NULL) {
        printf("please input the arg-N\n");
        return 0;
    }

    int N = atoi(argN); /*same to sscanf(argN, "%d", &N)*/
    if (N < 0 || N > 32) {
        printf("%d is to large!, the range of N is 0 to %d\n", N, 32);
        return 0;
    }

    // printf("%d\n", N);

    if(!free_wp(N)) {
        printf("No watchpoints number %d\n", N);
    }

    return 0;

}
```

- **info w 命令**：显示当前所有监视点。

添加到info命令中，输入info w时跳转至函数watchpoint_all_display()显示所有监视点。

```
static int cmd_info(char *args){
    switch(*args){
        case 'r':dum_regs();
        case 'w':watchpoint_all_display();
        return 0;
        default:
            return 1;
    }
}
```

watchpoint_all_display函数

```
void watchpoint_display(int N) {

    WP *wp = find_wp(N);
    if (wp == NULL) {
        printf("No breakpoint number %d\n", N);
        return;
    }

    printf("Num \t EXPR\n"); // TODO: complete this function
    printf("%d\t%s\n", wp->NO, wp->EXPR);

}
```

```
watchpoint 0: $eax
(nemu) si 5
100000:  b8 34 12 00 00          movl $0x1234,%eax
100005:  b9 27 00 10 00          movl $0x100027,%ecx
10000a:  89 01                  movl %eax, (%ecx)
10000c:  66 c7 41 04 01 00      movw $0x1,0x4(%ecx)
100012:  bb 02 00 00 00          movl $0x2,%ebx
(nemu)
```

```
For help, type "help"
(nemu) w $eip
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\$[a-z]{2,3}" at position 0 with len 4: $eip
watchpoint 0: $eip
(nemu) w $eax
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\$[a-z]{2,3}" at position 0 with len 4: $eax
watchpoint 1: $eax
(nemu) w $ecx
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\$[a-z]{2,3}" at position 0 with len 4: $ecx
watchpoint 2: $ecx
(nemu) info w
Num      EXPR
2        $ecx
1        $eax
0        $eip
(nemu) 2
Unknown command '2'
(nemu) d 2
NO 2 watchpoint has been deleted
(nemu) info w
No watchpoints
(nemu) info w
No watchpoints
(nemu) w $eax
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\$[a-z]{2,3}" at position 0 with len 4: $eax
watchpoint 2: $eax
```

任务4：实现监视点

在ui.c中的cmd_d和cmd_w中完成对应操作。

list_wp在watchpoint_all_display函数里实现，通过info w命令调用。

set_wp

```
WP *wp = new_wp(EXPR);
    if(wp != NULL)
        printf("watchpoint %d: %s\n", wp->NO, wp->EXPR);
```

del_wp, scan_wp

```
if(!free_wp(N)) {
    printf("No watchpoints number %d\n", N);
}
```

任务5：使用模拟断点

遇到的问题及解决办法

1.一开始在计算时会把16进制数当成10进制

解决办法：在rule中把10进制数的{"[0-9]+", TK_DEC}改为{"\\b[0-9]+\\b", TK_DEC}，用\\b表示开头和结尾，其中两个\\分别在vim和c中转义。

2.任务6中注意①：3*1+ (2+3) 出错

解决办法：还没解决，影响比较小。

3.实现p命令：p \$eax等较大的寄存器时，使用指针解引用会发生物理地址越界错误。

```
For help, type "help"
(nemu) p $eax
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\\${a-z}[2,3]" at position 0 with len 4: $eax
result = 0x1ee9cb6d 518638445
(nemu) p *$eax
[src/monitor/debug/expr.c,114,make_token] match rules[4] = "*" at position 0 with len 1: *
[src/monitor/debug/expr.c,114,make_token] match rules[6] = "\\${a-z}[2,3]" at position 1 with len 4: $eax
physical address(0x1ee9cb6d) is out of bound
nemu: src/memory/memory.c:15: paddr_read: Assertion 'addr < (128 * 1024 * 1024)' failed.
make: *** [Makefile:47: run] Aborted
tangxi@debian:~/ics2021/nemu$
```

解决办法：可能是因为创建虚拟机时选的是32位，舍友64位可以正常输出。

实验心得

通过这次的实验我学会了如何使用正则表达式和递归来实现表达式求值。用gdb调试程序和完善，实现监视点的功能。

其他备注

无