Lab 3

1. Functions

point	С	32bit	64bit
a. No args no ret	<pre>void func() { int x = 1, y = 2, z; z = x + y; } int main() { func(); }</pre>	func: .LFB0: .cfi_startproc endbr32 pushl %eb0 .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 subl \$16, %esp callx86.get_pc_thunk.ax addl \$_GLOBAL_OFFSET_TABLE_, %eax movl \$1, -12(%ebp) movl -12(%ebp), %edx movl -8(%ebp), %eax addl %edx, %eax movl %eax, -4(%ebp) nop leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc .LFE0: .size func,func .globl main .type main, @function main: .LFB1: .cfi_startproc endbr32 pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 callx86.get_pc_thunk.ax addl \$_GLOBAL_OFFSET_TABLE_, %eax call func movl \$0, %eax popl %ebp	.LFBO: .cfi_startproc endbr64 pushq %rbp .cfi_def_cfa_offset 16 .cfi_offset 6, -16 movq %rsp, %rbp .cfi_def_cfa_register 6 movl \$1, -12(%rbp) movl \$2, -8(%rbp), %edx movl -12(%rbp), %edx movl -8(%rbp), %eax addl %edx, %eax movl %eax, -4(%rbp) nop popq %rbp .cfi_def_cfa 7, 8 ret .cfi_endproc .LFEO: .size func,func .globl main .type main, @function main: .LFB1: .cfi_startproc endbr64 pushq %rbp .cfi_def_cfa_offset 16 .cfi_offset 6, -16 movq %rsp, %rbp .cfi_def_cfa_register 6 movl \$0, %eax call func movl \$0, %eax popq %rbp .cfi_def_cfa 7, 8 ret

```
int func()
b. No args,
                                                                                                                                       .cfi_startproc

endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
subl $16, %esp
call __x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
movl $1, -12(%ebp)
movl $2, -8(%ebp)
movl -12(%ebp), %edx
movl -8(%ebp), %eax
addl %edx, %eax
movl %eax, -4(%ebp)
                                                                                                                                                                                                                                                     .cfi_startproc
endbr64
                                                                      int x = 1, y = 2, z;
                                                                                                                                                                                                                                                 endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
movl $1, -12(%rbp)
movl $2, -8(%rbp)
movl -12(%rbp), %edx
movl -8(%rbp), %eax
%edx, %eax
ret
                                                                      z = x + y;
                                                                      return z;
                                                  int main()
                                                                      int a = func();
print("%d", a);
                                                                                                                                                                                                                                                                            %edx, %eax
%eax, -4(%rbp)
-4(%rbp), %eax
                                                                                                                                                          %eax, -4(%ebp)
-4(%ebp), %eax
                                                                                                                                                                                                                                                    movl
                                                                                                                                            movl
                                                                                                                                            leave
                                                                                                                                                                                                                                                    movl
                                                                                                                                            .cfi_restore 5
.cfi_def_cfa 4, 4
                                                                                                                                                                                                                                                                            %гЬр
                                                                                                                                                                                                                                                     popq
                                                                                                                                                                                                                                                     .cfi_def_cfa 7, 8
                                                                                                                                             .cfi_endproc
                                                                                                                                                                                                                                                     ret
                                                                                                                            .LFE0:
                                                                                                                                                                                                                                                     .cfi_endproc
                                                                                                                                            .size func, .-func
.section .rodata
                                                                                                                                                                                                                          .LFE0:
                                                                                                                                                                                                                                                     .size func, .-func
                                                                                                                           .I CO:
                                                                                                                                            .string "%d"
                                                                                                                                                                                                                                                     .section
                                                                                                                                                                                                                                                                                                     .rodata
                                                                                                                                            .text
.globl main
.type main, @function
                                                                                                                                                                                                                            .LC0:
                                                                                                                                                                                                                                                     .string "%d"
                                                                                                                                                                                                                                                     .text
.globl main
                                                                                                                           main:
.LFB1:
                                                                                                                                           .type
                                                                                                                                                                                                                                                                           main, @function
                                                                                                                                                                                                                          main:
                                                                                                                                                                                                                           .LFB1:
                                                                                                                                                                                                                                                     .cfi_startproc
                                                                                                                                                                                                                                                     endbr64
                                                                                                                                                                                                                                                   pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $16, %rsp
movl $0, %eax
                                                                                                                                            .cfi_
pushl
                                                                                                                                            pushl %ebx
pushl %ecx
                                                                                                                                            pushL %ecx
.cfi_escape 0xf,0x3,0x75,0x78,0x6
.cfi_escape 0x10,0x3,0x2,0x75,0x7c
subl $16, %esp
call __x86.get_pc_thunk.bx
addl $_GLOBAL_OFFSET_TABLE_, %ebx
call func
                                                                                                                                                                                                                                                    call
                                                                                                                                                                                                                                                                             func
                                                                                                                                                                                                                                                                                              -4(%rbp)
                                                                                                                                                                                                                                                     movl
                                                                                                                                                                                                                                                                            %eax.
                                                                                                                                                                                                                           func:
.LFB0:
                                                 int func(int d)
                                                                                                                                            .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
subl $16, %esp
call __x86.get_pc_thunk.ax
addl $_CLOBAL_OFFSET_TABLE_,
movl $2, -8(%ebp)
movl $(%ebp), %eax
imull -8(%ebp), %eax
movl %eax, -4(%ebp)
movl -4(%ebp), %eax
leave
.cfi_restore 5
c.1 arg
                                                                                                                                                                                                                                                  .cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
movl %edi, -20(%rbp)
movl $2, -8(%rbp)
rovl -20(%rbp), %eax
                                                                       int y = 2, z;
z = d * y;
                                                                        return z;
                                                                                                                                                                                                                                                   movl %edi, -20(%rbp)
movl $2, -8(%rbp)
movl -20(%rbp), %eax
imull -8(%rbp), %eax
movl %eax, -4(%rbp)
movl -4(%rbp), %eax
                                                 int main()
                                                                        int x = 5;
                                                                        int a = func(x);
printf("%d", a);
                                                                                                                                              .cfi_restore 5
.cfi_def_cfa 4, 4
ret
.cfi_endproc
                                                                                                                                                                                                                                                                            %гЬр
                                                                                                                                                                                                                                                    popq %rbp
.cfi_def_cfa 7, 8
                                                                                                                                                                                                                                                    ret ____
.cfi_endproc
                                                                                                                                              .size 1
                                                                                                                                                                                                                           .LFE0:
                                                                                                                                                                                                                                                                           func, .-func
n .rodata
                                                                                                                                                                                                                                                     .size :
                                                                                                                           .LCO:
                                                                                                                                              .string "%d"
.text
.globl main
.type main, @function
                                                                                                                                                                                                                           .LCO:
                                                                                                                                                                                                                                                     .string "%d"
                                                                                                                                                                                                                                                     .text
.globl main
.type main, @function
                                                                                                                            main:
.LFB1:
                                                                                                                                             .cfi_startproc
endbr32
leal 4(%esp), %ecx
.cfi_def_cfa 1, 0
andl $-16, %esp
pushl -4(%ecx)
pushl %ebp
movl %esp, %ebp
.cfi_escape 0x10,0x5,0x2,0x75,0
ushl %ebx
                                                                                                                                                                                                                           main:
.LFB1:
                                                                                                                                                                                                                                                    .cfi_startproc
endbr64
                                                                                                                                                                                                                                                  endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $16, %rsp
movl $5, -8(%rbp)
movl -8(%rbp), %eax
movl %eax, %edi
                                                                                                                                         movl %esp, %ebp
.cfi_escape 0x10,0x5,0x2,0x75,0
pushl %ebx
.cfi_escape 0xf,0x3,0x75,0x78,0
.cfi_escape 0x10,0x3,0x2,0x75,0
subl $16, %esp
call __x86.get_pc_thunk.bx
addl $_GLOBAL_OFFSET_TABLE_,
movl $5, -16(%ebp)
pushl -16(%ebp)
call func
                                                                                                                                                                                                                                                                             %eax, -4(%rbp)
                                                                                                                                                                                                                                                    movl
```

```
int func(int a, int b, int c)
                                                               func:
.LFB0:
d. many
                                                                                                                .LFB0:
                                  int res;
res = a * b + c;
return res;
                                                                       .cfi_startproc
                                                                                                                            .cfi_startproc
                                                                       endbr32
args
                                                                      pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
                                                                                                                            endbr64
                                                                                                                            pushq
                                                                                                                                        %гьр
                                                                               %esp, %ebp
                                                                                                                            .cfi_def_cfa_offset 16
                         int main()
                                                                       .cfi_def_cfa_register 5
subl $16, %esp
call __x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
                                                                                                                            .cfi_offset 6, -16
movq %rsp, %rbp
                                  int x = 5, y = 3, z = 7;
int a = func(x, y, z);
printf("%d", a);
                                                                       subl
                                                                                                                            movq
                                                                       call
                                                                                                                            .cfi_def_cfa_register 6
                                                                               8(%ebp), %eax
12(%ebp), %eax
                                                                       movl
                                                                                                                            movl
                                                                                                                                         %edi, -20(%rbp)
                                                                       imull
                                                                                                                                         %esi, -24(%rbp)
                                                                               %eax, %edx
16(%ebp), %eax
                                                                       movl
                                                                                                                                         %edx, -28(%rbp)
                                                                       movl
                                                                                                                            movl
                                                                       addl
                                                                               %edx, %eax
                                                                                                                                         -20(%rbp), %eax
                                                                                                                            movl
                                                                               %eax, -4(%ebp)
-4(%ebp), %eax
                                                                                     -4(%ebp)
                                                                       movl
                                                                                                                                         -24(%rbp),
                                                                                                                            imull
                                                                       leave
                                                                                                                            movl
                                                                                                                                         %eax, %edx
                                                                       .cfi_restore 5
.cfi_def_cfa 4, 4
                                                                                                                                         -28(%rbp), %eax
                                                                                                                                         %edx, %eax
                                                                                                                            addl
                                                                                                                                         %eax, -4(%rbp)
                                                                                                                            movl
                                                               main:
.LFB1:
                                                                                                                                         -4(%rbp), %eax
                                                                                                                            movl
                                                                       .cfi_startproc
                                                                                                                                         %гьр
                                                                                                                            popq
                                                                       endbr32
                                                                               4(%esp), %ecx
                                                                                                                            .cfi_def_cfa 7, 8
                                                                       .cfi_def_cfa 1, 0
andl $-16, %esp
pushl -4(%ecx)
                                                                                                               main:
                                                                       pushl
                                                                               %ebp
                                                                                                                .LFB1:
                                                                               %esp, %ebp
                                                                       movl
                                                                                                                            .cfi_startproc
                                                                            _escape 0x10,0x5,0x2,0x75,0
                                                                       .cfi
                                                                                                                            endbr64
                                                                       pushl
                                                                              %ecx
                                                                                                                            pushq
                                                                       .cfi_escape 0xf,0x3,0x75,0x78,0x6
.cfi_escape 0x10,0x3,0x2,0x75,0x7c
                                                                                                                                         %гьр
                                                                                                                            .cfi_def_cfa_offset 16
                                                                       subl
                                                                               $16, %esp
                                                                                                                            .cfi_offset 6, -16
                                                                                __x86.get_pc_thunk.bx
$_GLOBAL_OFFSET_TABLE_, %ebx
                                                                       call
                                                                       addl
                                                                                                                                         %rsp, %rbp
                                                                       movl
                                                                                                                            .cfi def cfa register 6
                                                                                    -20(%ebp)
                                                                       movl
                                                                                    -16(%ebp)
                                                                                                                                         $16, %rsp
                                                                                -16(%ebp)
                                                                       push1
                                                                                                                                         $5, -16(%rbp)
                                                                       pushl
                                                                                -24(%ebp)
                                                                                                                                         $3, -12(%rbp)
                                                                                                                                         $7, -8(%rbp)
                                                                                                                                         -8(%rbp), %edx
                                                                                                                                         -12(%rbp), %ecx
                                                                                                                                         -16(%rbp), %eax
                                                                                                                                         %ecx, %esi
                                                                                                                                         %eax, %edi
                                                                                                                                         func
                                                                                                                                         %eax, -4(%rbp)
```

General observations:

- register %eax is used as a return value
- 32bit and 64bit systems operate the setup and cleanup of stack frame differently(i.g. 32bit uses leave(equals movl + popl), 64bit operate with just popq
- another feature of 64bit system is that, unlike 32bit system, which allocate memory for local variables using, i.g., subl \$16, %esp in both main and leaf function, there is a "red zone" of 128 bytes below %rsp. These 128 bytes belong to

the function as long as it's a leaf function. Thus, all of local variables of a leaf function fit into the red zone, so no adjustment of %rsp needed (no instructions such as subq \$16, %rsp).

2. Local variables.

point	С	32bit	64bit
a. 1 loc var	int $x = 5$	<pre>pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 subl \$16, %esp callx86.get_pc_thunk.ax addl \$_GLOBAL_OFFSET_TABLE_, %eax movl \$5, -4(%ebp)</pre>	<pre>pushq %rbp .cfi_def_cfa_offset 16 .cfi_offset 6, -16 movq %rsp, %rbp .cfi_def_cfa_register 6 movl \$5, -4(%rbp)</pre>
b. 5 loc var	int a = 5, b = 4, c = -6, d = 8, i = 9;	<pre>subl \$32, %esp callx86.get_pc_thunk.ax addl \$_GLOBAL_OFFSET_TABLE_, %eax movl \$5, -20(%ebp) movl \$4, -16(%ebp) movl \$-6, -12(%ebp) movl \$8, -8(%ebp) movl \$9, -4(%ebp)</pre>	movl \$5, -20(%rbp) movl \$4, -16(%rbp) movl \$-6, -12(%rbp) movl \$8, -8(%rbp) movl \$9, -4(%rbp)
c. static array	int arr[50]; arr[7] = -345;	<pre>subl \$208, %esp callx86.get_pc_thunk.ax addl \$_GLOBAL_OFFSET_TABLE_, %eax movl \$-345, -172(%ebp)</pre>	Without stack protector
			subq \$88, %rsp movl \$-345, -180(%rbp)
			With stack protector
			movq %fs:40, %rax movq %rax, -8(%rbp)
			xorl %eax, %eax movl \$-345, -180(%rbp)
			movl \$0, %eax movq -8(%rbp), %rdx
			xorq %fs:40,´%rdx je .L3 callstack_chk_fail@PLT

```
int* p = (int*)malloc(sizeof(int)*10);
                                                                                         subq
                                                                                                    $16, %rsp
                                                            __x86.get_pc_thunk.bx
$_GLOBAL_OFFSET_TABLE_, %ebx
                                                    call
                          p[9] = 15;
                                                                                                    $40, %edi
                                                                                         novl
                                                    addl
                          free(p);
d. dynamic
                                                    subl
                                                            $12, %esp
                                                                                         call
                                                                                                    malloc@PLT
                                                    pushl
                                                                                                    %rax, -8(%rbp)
                                                                                         pvor
                                                           malloc@PLT
                                                    call
array(C)
                                                    addl
                                                            $16, %esp
                                                                                         pvom
                                                                                                    -8(%rbp), %rax
                                                            %eax, -12(%ebp)
                                                    movl
                                                                                         addq
                                                                                                    $36, %rax
                                                            -12(%ebp), %eax
                                                    movl
                                                            36, %eax
                                                    addl
                                                                                         movl
                                                                                                    $15, (%rax)
                                                    movl
                                                                (%eax)
                                                                                                    -8(%rbp), %rax
                                                    subl
                                                                                         pvom
                                                            $12, %esp
-12(%ebp)
                                                                %esp
                                                    pushl
                                                                                                    %rax, %rdi
                                                                                         pvor
                                                    call
                                                            free@PLT
                                                                                         call
                                                                                                    free@PLT
                                                    addl
                                                            $16, %esp
                                                    movl
                                                               %eax
                                                    leal
                                                            -8(%ebp)
                                                            $16, %esp
                           int *p = new int[10];
d. dynamic
                                                           __x86.get_pc_thunk.bx
$_GLOBAL_OFFSET_TABLE_, %ebx
                                                    call
                           p[8] = 15;
                                                    addl
                                                                                                    $16, %rsp
                                                                                         subq
                           delete[] p;
array(C++)
                                                    subl
                                                                                         movl
                                                                                                    $40, %edi
                                                    pushl
                                                           _Znaj@PLT
$16, %esp
                                                    call
                                                                                         call
                                                                                                     _Znam@PLT
                                                    addl
                                                                                         movq
                                                                                                    %rax, -8(%rbp)
                                                           %eax, -12(%ebp)
                                                    movl
                                                            -12(%ebp), %eax
                                                    movl
                                                                                                    -8(%rbp), %rax
                                                                                         movq
                                                    addl
                                                            $32, %eax
                                                                                                    $32, %rax
                                                                                         addq
                                                    movl
                                                            $15, (%eax)
                                                               -12(%ebp)
                                                    cmpl
                                                                                         movl
                                                                                                    $15, (%rax)
                                                    jе
                                                                                                    $0, -8(%rbp)
                                                                                         cmpq
                                                    subl
                                                           $12, %esp
-12(%ebp)
                                                    pushl
                                                                                                    .L2
                                                                                         je
                                                    call
                                                            ZdaPv@PLT
                                                                                         movq
                                                                                                    -8(%rbp), %rax
                                                    addl
                                                                                                    %rax, %rdi
                                                                                         pvom
                                                                                                     ZdaPv@PLT
                                                                                         Znam@PLT = new[]
                                                                                         ZdaPV@PLT =
                                                                                        delete[]
```

- static array in 64bit shift %rsp to a less amount, than the size of array, perhaps, because of the red zone.
- In 32bit system size of dynamic array is pushed in stack; in 64bit is moved to %edi
- %eax/%rax initially refers to the first element of dynamic array(it must be so because after we call malloc/new[], the result of function, i.e. pointer to the first element, is written in %eax)

3. Structures

point	C++	32bit	64bit
b. global struct	<pre>struct A{</pre>	.align 4 .type s1, @object .size s1, 16 s1: .zero 16 .text .globl main .type main, @function main: .LFB0: .cfi_startproc endbr32 pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 callx86.get_pc_thunk.ax addl \$_GLOBAL_OFFSET_TABLE_, movl \$7, s1@GOTOFF(%eax) fldl _LCO@GOTOFF(%eax) fstpl 4+s1@GOTOFF(%eax) movb \$114, 12+s1@GOTOFF(%eax)	.align 16 .type s1, @object .size s1, 24 s1: .zero 24 .text .globl main .type main, @function main: .LFB0: .cfi_startproc endbr64 pushq %rbp .cfi_def_cfa_offset 16 .cfi_offset 6, -16 movq %rsp, %rbp .cfi_def_cfa_register 6 movl \$7, s1(%rip) movsd .LCO(%rip), %xmm0 movsd %xmm0, 8+s1(%rip) movb \$114, 16+s1(%rip)
c. static array as a member	<pre>struct A{</pre>	.zero 28 .text .globl main .type main, @function main: .LFB0: .cfi_startproc endbr32 pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 callx86.get_pc_thunk.ax addl \$_GLOBAL_OFFSET_TABLE_, movl \$7, si@GOTOFF(%eax) fstpl 4+si@GOTOFF(%eax) fstpl 4+si@GOTOFF(%eax) movb \$114, 12+si@GOTOFF(%eax) movl \$5, 16+si@GOTOFF(%eax) movl \$2, 20+si@GOTOFF(%eax) movl \$-4, 24+si@GOTOFF(%eax)	<pre>s1: .zero 32 .text .globl main .type main, @function main: .LFB0: .cfi_startproc endbr64 pushq %rbp .cfi_def_cfa_offset 16 .cfi_offset 6, -16 movq %rsp, %rbp .cfi_def_cfa_register 6 movl \$7, \$1(%rip) movsd .LC0(%rip), %xmm0 movsd %xmm0, 8+s1(%rip) movb \$114, 16+s1(%rip) movl \$5, 20+s1(%rip) movl \$2, 24+s1(%rip) movl \$-4, 28+s1(%rip)</pre>

```
_Z4funcR1A
_Z4funcR1A, @function
                                                                                                                                                                                                                                      .globl _Z4funcR1A
                                                         struct A{
d. struct as an
                                                                                                                                                                                                                                       .type
                                                                                                                                                                                                                                                     _Z4funcR1A, @function
                                                                                                                                      Z4funcR1A:
                                                                                   int a;
                                                                                                                                                                                                                        Z4funcR1A:
arg in function
                                                                                                                                                  .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
                                                                                   double b;
                                                                                                                                                                                                                                       .cfi_startproc
                                                                                                                                                                                                                                 endbr64
                                                                                   char c;
                                                                                   int arr[3];
                                                                                                                                                 movl %esp, %ebp
.cfi_def_cfa_register 5
call _x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
movl 8(%ebp), %eax
movl $(%ebp), %eax
movl $(%ebp), %eax
movl $(%ebp), %eax
movl 8(%ebp), %eax
movl $(%ebp), %eax
movl $90, 24(%eax)
                                                          };
                                                           void func(A &s)
                                                                                                                                                                                                                                                    $113, 16(%rax)
-8(%rbp), %rax
$90, 28(%rax)
                                                                                   s.a = 2;
                                                                                  s.c = 'q^{\prime\prime};
                                                                                                                                                  nop
popl
                                                                                                                                                                                                                                      movl
                                                                                                                                                               %ebp
                                                                                                                                                                                                                                      nop
                                                                                   s.arr[2] = 90;
                                                                                                                                                   .cfi_restore 5
.cfi_def_cfa 4, 4
                                                                                                                                                                                                                                      popq %rbp
.cfi_def_cfa 7, 8
                                                                                                                                                                                                                                      popq
                                                                                                                                                  ret
.cfi_endproc
                                                                                                                                      .LFE0:
                                                                                                                                                                                                                      .LFE0:
                                                                                                                                                   .size _Z4funcR1A, .-_Z4funcR1A
.globl main
.type main, @function
                                                          int main()
                                                                                                                                                                                                                                                      _Z4funcR1A, .-_Z4funcR1A
                                                                                                                                                                                                                                      .size
                                                                                                                                                                                                                                      .stze __Z4TunckiA, .-_/
.globl main
.type main, @function
                                                                                                                                     main:
.LFB1:
                                                                                   A s1;
                                                                                                                                                 .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
subl $32, %esp
call __x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
movb $116, -10(%ebp)
leal -28(%ebp), %eax
pushl %eax
                                                                                                                                                                                                                      main:
                                                                                                                                                                                                                       .LFB1:
                                                                                   s1.c = 't';
                                                                                                                                                                                                                                      .cfi_startproc
                                                                                   func(s1);
                                                                                                                                                                                                                                     endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $32, %rsp
movb $116, -16(%rbp)
leaq -32(%rbp), %rax
movq %rax, %rdi
call _Z4funcR1A
                                                                                                                                                                _Z4funcR1A
$4, %esp
                                                                                                                                                                _Z4funcv
_Z4funcv, @function
                                                          struct A{
                                                                                                                                                                                                                                        .globl _Z4funcv
e. struct as a
                                                                                                                                                                                                                                                        _Z4funcv, @function
                                                                                                                                                                                                                                        .type
                                                                                    int a;
                                                                                                                                      _Z4funcv:
                                                                                                                                                                                                                       _Z4funcv:
                                                                                                                                                  .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
call __x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
movl 8(%ebp). %eax
                                                                                   double b:
return value
                                                                                                                                                                                                                        .LFB0:
                                                                                                                                                                                                                                        .cfi startproc
                                                                                   char c;
                                                                                                                                                                                                                                      endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
fa_register
                                                                                   int arr[3];
                                                          };
                                                                                                                                                                                                                                       movq %rsp, %rbp
.cfi_def_cfa_register 6
movq %rdi, -8(%rbp)
movq -8(%rbp), %rax
movl $2, (%rax)
movq -8(%rbp), %rax
movb $113, 16(%rax)
movq -8(%rbp), %rax
movl $90, 28(%rax)
                                                          A func()
                                                                                                                                                               $_GLOBAL_OFFSE1
8(%ebp), %eax
$2, (%eax)
8(%ebp), %eax
$113, 12(%eax)
8(%ebp), %eax
$90, 24(%eax)
                                                                                   A s;
                                                                                                                                                   movb
                                                                                   s.a = 2;
                                                                                                                                                   movl
                                                                                   s.c = 'q';
s.arr[2] = 90;
                                                                                                                                                   nop
movl
                                                                                                                                                                 8(%ebp), %eax
                                                                                                                                                    .cfi restore
                                                                                                                                                                                                                                       nop
                                                                                                                                                     .cfi_def_cfa 4, 4
ret $4
                                                                                   return s:
                                                                                                                                                                                                                                                        -8(%rbp), %rax
                                                                                                                                                                                                                                       movq
                                                                                                                                                   ret $4
.cfi_endproc
                                                                                                                                                                                                                                                        %гьр
                                                                                                                                                                                                                                       popq
                                                                                                                                                                                                                                       .cfi_def_cfa 7, 8
                                                                                                                                      .LFE0:
                                                                                                                                                    .size _Z4funcv, .-_Z4funcv
.globl main
                                                                                                                                                                                                                                       ret
                                                                                                                                                                                                                                        .cfi_endproc
                                                           int main()
                                                                                                                                                                main, @function
                                                                                                                                                                                                                       .LFE0:
                                                                                                                                     main:
.LFB1:
                                                                                                                                                                                                                                       .size _Z4funcv, .-_Z4funcv
.globl main
.type main, @function
                                                                                                                                                  .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
subl $32, %esp
call _x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
leal -28(%ebp), %eax
pushl %eax
                                                                                   A s1 = func();
                                                                                                                                                                                                                      main:
                                                                                                                                                                                                                       .LFB1:
                                                                                                                                                                                                                                       .cfi_startproc
endbr64
                                                                                                                                                                                                                                       pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
                                                                                                                                                                                                                                       .cfi_def_cfa_register 6
subq $32, %rsp
leaq -32(%rbp), %rax
movq %rax, %rdi
                                                                                                                                                                   74funcy
                                                                                                                                                                                                                                                          Z4funcv
```

• in 32bit system Global offset table is used to dynamically access structures(in this task). In a. and b. @GOTOFF is used for global structures – perhaps it stands for GOT Offset, meaning making an offset from GOT address. In 64bit system register %rip was responsible for that.

4. Pointers and references

```
64bit
           point
                                                       C++
                                                                                                       32bit
                                                                                     _Z4func1A:
.LFB0:
                                                                                                                                                   .globl
.type
                                     struct A{
a. struct
                                                                                                                                                             _Z4func1A, @function
                                                                                              .cfi_startproc
endbr32
                                                     int a;
                                                                                                                                         Z4func1A:
                                                                                                                                         LFB0:
                                                     double b;
                                                                                                                                                   .cfi_startproc
                                                                                                                                                  endbr64
                                                     char c;
                                                                                                                                                  pushq %rbp
.cfi_def_cfa_offset 16
                                                                                                                                                  pushq
                                                     int arr[3];
                                                                                                       [_cfa_register 5
    __x86.get_pc_thunk.ax
$_GLOBAL_OFFSET_TABLE_, %eax
                                                                                                                                                   .cfi_der_cro_c
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
movl 16(%rbp), %eax
                                     void func(A s)
                                                                                                                                                  addl
                                                                                                                                                             $2, %eax
%eax, 16(%rbp)
$113, 32(%rbp)
                                                                                                                                                  movl
                                                                                                                                                  movb
                                                                                                                                                             $90, 36(%rbp)
                                                                                                                                                  movl
                                                                                                                                                             40(%rbp), %eax
                                                     s.arr[0] = 90;
                                                                                                                                                  movl
                                                                                                                                                             %eax, 40(%rbp)
                                                                                              ret .cfi_endproc
                                                                                                                                                   .cfi_def_cfa 7, 8
                                                                                     .LFE0:
                                                                                                       _Z4func1A, .-_Z4func1A
                                                                                                                                                   .cfi_endproc
                                     int main()
                                                                                                                                         .LFE0:
                                                                                                      main, @function
                                                                                     main:
.LFB1:
                                                                                                                                                              _Z4func1A, .-_Z4func1A
                                                                                                                                                   .size
                                                                                                                                                  .globl main
.type main
                                                                                              .cfi_startproc
                                                        s1;
                                                                                                                                                            main, @function
                                                                                                                                        main:
                                                                                                                                                   .cfi_startproc
                                                          arr[1] = 9;
                                                                                                                                                  pushq %rbp
.cfi_def_cfa_offset 16
                                                                                                                                                   .cfi_offset 6, -16
novq %rsp, %rbp
                                                                                                           -28<mark>(%ebp)</mark>
9, -16(%ebp)
                                                                                                                                                   .cfi_def_cfa_register 6
                                                                                                                                                                  %rsp
-32(%rbp)
                                                                                                                                                  suba
                                                                                                                                                  movl
                                                                                                                                                  movb
                                                                                                                                                             $119. -16(%rbp)
                                                                                                                                                                   -8(%rbp)
                                                                                                                                                  movl
                                                                                                                                                  pushq
                                                                                                                                                  pushq
```

```
struct A{
                                                                                                                                                                                                                         _Z4funcP1A:
                                                                                                                                                                                                                                                                                                                                                              _Z4funcP1A:
                                                                                                                                                                                                                                             .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
call _x86.get_pc_thunk.ax
addl $\, GLOBAL_OFFSET_TABLE_\, %eax
movl &(%eax), %eax
movl &(%eax), %edx
movl &(%ebp), %eax
movl &(%eax), %edx
movl &(%ebp), %eax
b. struct
                                                                                                                                int a;
                                                                                                                                                                                                                                                                                                                                                                                      .cfi_startproc
                                                                                                                                 double b;
                                                                                                                                                                                                                                                                                                                                                                                   endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
movq %rdi, -8(%rbp), %rax
movl (%rax), %eax
leal 2(%rax), %edx
movq -8(%rbp), %rax
movl %edx, (%rax)
movq -8(%rbp), %rax
movl %edx, (%rax)
movq -8(%rbp), %rax
movl $113, 16(%rax)
movq -8(%rbp), %rax
movl $90, 20(%rax)
movq -8(%rbp), %rax
movl 24(%rax), %eax
leal 1(%rax), %edx
movq -8(%rbp), %rax
movl 24(%rax), %eax
leal 1(%rax), %edx
movq -8(%rbp), %rax
movl %edx, 24(%rax)
nop
                                                                                                                                                                                                                                                                                                                                                                                     endbr64
pointer
                                                                                                                                 char c;
                                                                                                                                  int arr[3];
                                                                                               };
                                                                                               void func(A *s)
                                                                                                                                 s->a += 2;
                                                                                                                                s->c = 'q';
s->arr[0] = 90;
s->arr[1] += 1;
                                                                                               int main()
                                                                                                                                                                                                                                               nop
popl
                                                                                                                                  A s1;
                                                                                                                                                                                                                                                popl %ebp
.cfi_restore 5
.cfi_def_cfa 4, 4
                                                                                                                                  s1.a = 4;
                                                                                                                                                                                                                                                                                                                                                                                     popq %rbp
.cfi_def_cfa 7, 8
                                                                                                                                  s1.c = 'W';
                                                                                                                                                                                                                                              ret
.cfi_endproc
                                                                                                                                 s1.arr[1] = 9;
                                                                                                                                                                                                                                                                                                                                                                                      .cfi_endproc
                                                                                                                                                                                                                        .LFE0:
                                                                                                                                 func(&s1);
                                                                                                                                                                                                                                                                                                                                                             .LFE0:
                                                                                                                                                                                                                                                .size _Z4funcP1A, .-_Z4funcP1A
.section .rodata
                                                                                                                                                                                                                                                                                                                                                                                      .size _Z4funcP1A, .-_Z4funcP1A
.section .rodata
                                                                                                                                  printf("%d", s1.a);
                                                                                                                                                                                                                                                .string "%d"
                                                                                                                                                                                                                                                                                                                                                             .LC0:
                                                                                                                                                                                                                                                .text
.globl main
.type main, @function
                                                                                                                                                                                                                                                                                                                                                                                      .text
.globl main
.type main, @function
                                                                                                                                                                                                                         main:
                                                                                                                                                                                                                                                                                                                                                             main:
                                                                                                                                                                                                                                                .cfi_startproc
                                                                                                                                                                                                                                           .cfi_startproc
endbr32
leal 4(%esp), %ecx
.cfi_def_cfa 1, 0
andl $-16, %esp
pushl -4(%ecx)
pushl %ebp
movl %esp, %ebp
.cfi_escape 0x10,0x5,0x2,0x75,0
pushl %ebx
pushl %ebx
                                                                                                                                                                                                                                                                                                                                                             .LFB1:
                                                                                                                                                                                                                                                                                                                                                                                      .cfi_startproc
                                                                                                                                                                                                                                                                                                                                                                                    endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $32, %rsp
movl $4, -32(%rbp)
movb $119, -16(%rbp)
movl $9, -8(%rbp)
leaq -32(%rbp), %rax
movq %rax, %rdi
call _Z4funcP1A
                                                                                                                                                                                                                                                                                                                                                                                      endbr64
                                                                                                                                                                                                                                                                                                                                                                                   movb
                                                                                                                                                                                                                                                                    $32, %esp

_x86.get_pc_thunk.bx

$\sum_c108AL_OFFSET_TABLE_, %ebx

$4, -36(%ebp)

$19, -24(%ebp)

$9, -16(%ebp)

-36(%ebp), %eax

%eav
                                                                                                                                                                                                                                                call
                                                                                                                                                                                                                                                addl
                                                                                                                                                                                                                                                movl
                                                                                                                                                                                                                                                                                                                                                                                                             _Z4funcP1A
                                                                                                                                                                                                                                                                                                                                                                                                                -32(%rbp), %eax
                                                                                                                                                                                                                                                movl
leal
                                                                                                                                                                                                                                               pushl
call
                                                                                                                                                                                                                                                                     %eax
                                                                                                                                                                                                                                                                     _Z4funcP1A
$4, %esp
-36(%ebp), %eax
$8, %esp
                                                                                                                                                                                                                                                addl
```

```
c. struct
                                                                                                             int a;
                                                                                                                                                                                                         .cfi_startproc
endbr32
pushl %ebp
                                                                                                                                                                                                                                                                                                                         .cfi_startproc
                                                                                                            double b;
                                                                                                                                                                                                                                                                                                                        endbr64
reference
                                                                                                                                                                                                                                                                                                                       endbro4
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
                                                                                                                                                                                                        pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 call __x86.get_pc_thunk.ax addl $_GLOBAL_OFFSET_TABLE_, %eax movl &(%eax), %eax leal 2(%eax), %edx movl &(%ebp), %eax movl &(%ebp), %eax
                                                                                                            char c;
                                                                                                            int arr[3];
                                                                                                                                                                                                                                                                                                                       .cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
movq %rdi, -8(%rbp)
movq -8(%rbp), %rax
movl (%rax), %eax
leal 2(%rax), %edx
movq -8(%rbp), %rax
movl %edx, (%rax)
movq -8(%rbp), %rax
movb $113, 16(%rax)
movq -8(%rbp), %rax
movl $90, 20(%rax)
movq -8(%rbp), %rax
movl 24(%rax), %eax
leal 1(%rax), %edx
movq -8(%rbp), %rax
movl & %edx, 24(%rax)
nop
                                                                                void func(A &s)
                                                                                                            s.a += 2;
                                                                                                           s.c = 'q';
s.arr[0] = 90;
s.arr[1] += 1;
                                                                                                                                                                                                                           $90, 16(%eax)

8(%ebp), %eax

20(%eax), %eax

1(%eax), %edx

8(%ebp), %eax

%edx, 20(%eax)
                                                                                                                                                                                                          leal
                                                                                int main()
                                                                                                                                                                                                                          %ebp
                                                                                                                                                                                                                                                                                                                        пор
                                                                                                             s1.a = 4;
                                                                                                                                                                                                         .cfi_restore 5
.cfi_def_cfa 4, 4
                                                                                                                                                                                                                                                                                                                       popq %rbp
.cfi_def_cfa 7, 8
                                                                                                            s1.c = 'W';
                                                                                                                                                                                                         ret
.cfi_endproc
                                                                                                            s1.arr[1] = 9;
                                                                                                                                                                                                                                                                                                                        ret
.cfi endproc
                                                                                                                                                                                       .LFE0:
                                                                                                            func(s1);
                                                                                                                                                                                                         .size _Z4funcR1A, .-_Z4funcR1A
.section .rodata
                                                                                                                                                                                                                                                                                                   .LFE0:
                                                                                                            printf("%d", s1.a);
                                                                                                                                                                                                                                                                                                                                           _Z4funcR1A, .-_Z4funcR1A
                                                                                                                                                                                                                                                                                                                         .section
                                                                                                                                                                                                                                                                                                                                                                  .rodata
                                                                                                                                                                                                         .text
.globl main
                                                                                                                                                                                                                                                                                                                        .string "%d"
                                                                                                                                                                                                                                                                                                                        .text
.globl main
                                                                                                                                                                                                          .type main, @function
                                                                                                                                                                                      main:
.LFB1:
                                                                                                                                                                                                                                                                                                                                          main, @function
                                                                                                                                                                                                                                                                                                                         .tvpe
                                                                                                                                                                                                                                                                                                   main:
                                                                                                                                                                                                         .cfi_startproc
endbr32
                                                                                                                                                                                                                                                                                                                       .cfi_startproc
endbr64
                                                                                                                                                                                                                           4(%esp), %ecx
                                                                                                                                                                                                         leal 4(%esp), %cfi_def_cfa 1, 0
andl $-16, %esp
pushl -4(%ecx)
pushl %esp
                                                                                                                                                                                                                                                                                                                      endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
                                                                                                                                                                                                                           %esp, %ebp
cape 0x10,0x5,0x2,0x75,0
                                                                                                                                                                                                                                                                                                                       movq %rsp, %rbp
.cfi_def_cfa_register 6
subq $32, %rsp
movl $4, -32(%rbp)
movb $119, -16(%rbp)
movl $9, -8(%rbp)
leaq -32(%rbp), %rax
movq %rax, %rdi
call _Z4funcR1A
                                                                                                                                                                                                         .cfi_escape 0
pushl %ebx
pushl %ecx
                                                                                                                                                                                                                           $32, %esp

_x86.get_pc_thunk.bx

$\sum_GLOBAL_OFFSET_TABLE_, %ebx

$4, -36(%ebp)

$119, -24(%ebp)

$9, -16(%ebp)

-36(%ebp), %eax
                                                                                                                                                                                                                                                                                                                      movb
                                                                                                                                                                                                         call
addl
                                                                                                                                                                                                          leal
                                                                                                                                                                                                                          %eax
_Z4funcR1A
$4, %esp
-36(%ebp), %eax
                                                                                                                                                                                                                            S8. %esp
```

• in a. all struct members are pushed into stack; in b. and c. (pointers and references) struct members are not pushed into stack, only registers and stack frame(via i(%rbp)) are used.

5. Heavy structures

```
C++
                                                                                                                                                                    32bit
                                                                                                                                                                                                                                                      64bit
                  point
                                                                                                                                      _Z4func1A:
                                                            const int n = 10000000;
                                                           const the
struct A{
    int arr1[n];
    acc2[n];
                                                                                                                                                                                                                         Z4func1A:
a. struct as an
                                                                                                                                                   .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
subl $16, %esp
call __x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
movl $0, -4(%ebp)
                                                                                                                                                                                                                          .LFB0:
                                                                                                                                                                                                                                          .cfi_startproc
                                                                           int arr2[n];
                                                                                                                                                                                                                                         endbr64
arg
                                                                                                                                                                                                                                         endbro4
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
                                                           }:
                                                            void func(A s)
                                                                                                                                                                                                                                         .cfi_def_cfa_register 6
movl $0, -4(%rbp)
                                                                                           s.arr1[i] = i - 1;
s.arr2[i] = i;
                                                                                                                                                                                                                         .L3:
                                                                                                                                      .L3:
                                                                                                                                                                                                                                                          $9999999, -4(%rbp)
                                                                                                                                                                                                                                         cmpl
                                                                                                                                                    cmpl
                                                                                                                                                                  $9999999, -4(%ebp)
                                                                                                                                                                 $9999999, -4(%ebp)
.L4
-4(%ebp), %eax
-1(%eax), %edx
-4(%ebp), %eax
%edx, 8(%ebp,%eax,4)
-4(%ebp), %eax
10000000(%eax), %edx
-4(%ebp), %eax
%eax, 8(%ebp,%edx,4)
$1, -4(%ebp)
.L3
                                                                                                                                                                                                                                         jg
movl
                                                                                                                                                                                                                                                         -4(%rbp), %eax
-1(%rax), %edx
-4(%rbp), %eax
                                                                                                                                                    jg
movl
leal
                                                                                                                                                                                                                                         leal
                                                            int main()
                                                                                                                                                    movl
movl
                                                                                                                                                                                                                                         movl
                                                                                                                                                                                                                                         cltq
                                                                           A s1;
func(s1);
                                                                                                                                                    movl
leal
                                                                                                                                                                                                                                                         %edx, 16(%rbp,%rax,4)
-4(%rbp), %eax
                                                                                                                                                                                                                                         movĺ
                                                                                                                                                                                                                                         movl
                                                                                                                                                    movl
                                                                                                                                                                                                                                         cltq
                                                                                                                                                                                                                                         leaq
                                                                                                                                                                                                                                                          10000000(%rax), %rdx
                                                                                                                                                     addl
                                                                                                                                                                                                                                                         -4(%rbp), %eax
%eax, 16(%rbp,%rdx,4)
$1, -4(%rbp)
                                                                                                                                                                                                                                         movl
                                                                                                                                                                                                                                         movl
                                                                                                                                                    nop
leave
                                                                                                                                                                                                                                         addl
                                                                                                                                                                                                                                         jmp
                                                                                                                                                    .cfi_restore 5
.cfi_def_cfa 4, 4
                                                                                                                                                    ret
.cfi endproc
                                                                                                                                                                                                                                         nop
                                                                                                                                                                                                                                         popq %rbp
.cfi_def_cfa 7, 8
                                                                                                                                                                                                                                                         %гьр
                                                                                                                                        LFE0:
                                                                                                                                                     .size _Z4func1A, .-_Z4func1A
.globl main
.type main, @function
                                                                                                                                                                                                                                         .cfi_endproc
                                                                                                                                                                                                                          .LFE0:
                                                                                                                                      main:
.LFB1:
                                                                                                                                                                                                                                         .size _Z4func1A, .-_Z4func1A
.globl main
                                                                                                                                                    .cfi_startproc
endbr32
leal 4(%esp), %ecx
                                                                                                                                                                                                                                          .type main, @function
                                                                                                                                                  endbr32
leal 4(%esp), %ecx
.cfi_def_cfa 1, 0
andl $-16, %esp
pushl -4(%ecx)
pushl %ebp
movl %esp, %ebp
.cfi_escape 0x10,0x5,0x2,0x75,0
pushl %ebx
pushl %ex
                                                                                                                                                                                                                         main:
                                                                                                                                                                                                                         .LFB1:
                                                                                                                                                                                                                                         .cfi_startproc
endbr64
                                                                                                                                                                                                                                         endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq -79998976(%rsp), %r11
                                                                                                                                                    pushl
                                                                                                                                                                 %ecx
                                                                                                                                                                    -79998976(%esp), %eax
                                                                                                                                                     leal
                                                                                                                                        .LPSRL0:
                                                                                                                                                                                                                          .LPSRL0:
                                                                                                                                                                   $4096, %esp
$0, (%esp)
                                                                                                                                                                                                                                                         $4096, %rsp
$0, (%rsp)
%r11, %rsp
.LPSRL0
                                                                                                                                                                                                                                         subq
                                                                                                                                                     orl
                                                                                                                                                     cmpl
                                                                                                                                                                   %eax, %esp
.LPSRL0
                                                                                                                                                                                                                                         ога
                                                                                                                                                     jne
                                                                                                                                                                                                                                       cmpq
                                                                                                                                                                  LPSRL0
$1024, %esp
_x86.get_pc_thunk.ax
$_GLOBAL_OFFSET_TABLE_, %eax
$80000000, %esp
%esp, %edx
%edx, %ecx
-80000008(%ebp), %edx
$80000000, %ebx
$4, %esp
                                                                                                                                                     subl
call
                                                                                                                                                                                                                                         jne
                                                                                                                                                                                                                                                         $1024, %rsp
$80000000, %rsp
                                                                                                                                                                                                                                         subq
                                                                                                                                                     addl
subl
                                                                                                                                                                                                                                         subq
                                                                                                                                                                                                                                                         %rsp, %rax
%rax, %rcx
                                                                                                                                                                                                                                         pvom
                                                                                                                                                                                                                                         movq
                                                                                                                                                     movl
                                                                                                                                                     leal
                                                                                                                                                                                                                                                           -80000000(%rbp),
                                                                                                                                                                                                                                         leaq
                                                                                                                                                                                                                                                                                              %гах
                                                                                                                                                     movl
                                                                                                                                                                                                                                                          $80000000, %edx
%rax, %rsi
%rcx, %rdi
                                                                                                                                                                   $4, %esp
%ebx
                                                                                                                                                     subl
                                                                                                                                                                                                                                         movl
                                                                                                                                                     pushl
                                                                                                                                                                                                                                         movq
                                                                                                                                                     pushl
                                                                                                                                                                                                                                         movq
                                                                                                                                                     pushl
movl
call
                                                                                                                                                                   %ecx
                                                                                                                                                                                                                                         call
                                                                                                                                                                                                                                                          memcpy@PLT
                                                                                                                                                                   %eax, %ebx
memcpy@PLT
$16, %esp
                                                                                                                                                                                                                                                            Z4func1A
                                                                                                                                                                                                                                         call
                                                                                                                                                                                                                                                           $80000000.
                                                                                                                                                                                                                                                                                  %гѕр
                                                                                                                                                                   $16, %esp
Z4func1A
```

```
b. struct as a return value
```

```
const int n = 100000;
                                                                        Z4funcv:
                                                                                                                                                       _Z4funcv:
                                                                                .cfi_startproe
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
subl $16, %esp
call _x86.get_pc_thunk.ax
addl $_GLOBAL_OFFSET_TABLE_, %eax
movl $0, -4(%ebp)
 struct A{
                int arr1[n];
                                                                                                                                                                       .cfi_startproc
               int arr2[n];
                                                                                                                                                                       endbr64
                                                                                                                                                                    pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
};
A func()
               A s;
for(int i = 0; i < n; i ++)
                                                                                                                                                                       .cfi_def_cfa_register 6
movq %rdi, -24(%rbp)
movl $0, -4(%rbp)
                              s.arr1[i] = i - 1;
s.arr2[i] = i;
                                                                     .L3:
                                                                                                                                                                      $99999, -4(%rbp)
                                                                                                                                                                       cmpl
               }
return s;
                                                                                     jg
movl
leal
                                                                                                  .L5
-4(%ebp), %eax
-1(%eax), %ecx
8(%ebp), %eax
-4(%ebp), %edx
%ecx, (%eax,%edx,4)
8(%ebp), %edx
100000(%edx), %ecx
-4(%ebp), %edx
%edx, (%eax,%ecx,4)
$1, -4(%ebp)
.L3
                                                                                     movl
movl
int main()
                                                                                     movl
                                                                                     movl
              A s1;
s1 = func();
printf("%d", s1.arr2[1090]);
                                                                                     movl
leal
                                                                                     movl
movl
                                                                                      add1
                                                                                     jmp
                                                                        .L5:
                                                                                     пор
                                                                                     movl
leave
                                                                                                8(%ebp), %eax
                                                                                     .cfi_restore 5
.cfi_def_cfa 4, 4
ret $4
                                                                                                                                                                       nop
                                                                                                                                                                                       -24(%rbp), %rax
                                                                                                                                                                       movq
                                                                          .LFB1:
                                                                                                                                                                                      %гЬр
                                                                                                                                                                       popq
                                                                                       .cfi_startproc
                                                                                   .cfi_starcp.
endbr32
leal 4(%esp), %ecx
.cfi_def_cfa 1, 0
andl $-16, %esp
pushl -4(%ecx)
pushl %ebp
%esp, %ebp
                                                                                                                                                                        .cfi_def_cfa 7, 8
                                                                                                                                                                       ret
.cfi_endproc
                                                                                                                                                       .LFE0:
                                                                                                                                                                       .size
                                                                                                                                                                                      _Z4funcv . . _Z4funcv
                                                                                                                                                                                                       .rodata
                                                                                                                                                                       .section
                                                                                       movl %esp, %ebp
.cfi_escape 0x10,0x5,0x2,0x75,0
                                                                                                                                                       .LC0:
                                                                                                                                                                       .string "%d"
                                                                                      pushl %ebx
pushl %ecx
                                                                                                                                                                       .text
                                                                                                                                                                       .globl main
                                                                                       .cfi_escape 0xf,0x3,0x75,0x78,0x6
.cfi_escape 0x10,0x3,0x2,0x75,0x7c
leal -1597440(%esp), %eax
                                                                                                                                                                       .type main, @function
                                                                                      leal
                                                                                                                                                       main:
                                                                         .LPSRL0:
                                                                                                                                                       .LFB1:
                                                                                                   $4096, %esp
$0, (%esp)
%eax, %esp
.LPSRL0
                                                                                      subl
                                                                                                                                                                       .cfi_startproc
                                                                                      orl
                                                                                      cmpl
jne
                                                                                                                                                                       endbr64
                                                                                                                                                                      endbrb4
pushd %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq -1597440(%rsp), %r11,
                                                                                                   .LPSKLU
$2560, %esp
_x86.get_pc_thunk.bx
$_GLOBAL_OFFSET_TABLE_, %ebx
-1600008(%ebp), %eax
                                                                                      subl
call
                                                                                      addl
leal
                                                                                      pushl
call
                                                                                                   _Z4funcv
-800008(%ebp), %eax
-1600008(%ebp), %edx
                                                                                      leal
leal
                                                                                                                                                                                        $4096, %rsp
$0, (%rsp)
%r11, %rsp
.LPSRL0
                                                                                                                                                                        subq
                                                                                      movl
subl
                                                                                                   $800000, %ecx
$4, %esp
%ecx
                                                                                                                                                                        ога
                                                                                                                                                                        cmpq
                                                                                                                                                                        jne
                                                                                                    %edx
                                                                                       pushl
                                                                                                                                                                                        $2560, %rsp
-1600000(%rbp), %rax
                                                                                                                                                                        subq
                                                                                      pushl
call
                                                                                                    memcpy@PLT
$16, %esp
-395648(%ebp),
                                                                                                                                                                        leaq
                                                                                       addl
                                                                                                                                                                                        %rax, %rdi
                                                                                                                                                                       movq
                                                                                                                                                                                        Z4funcv
-800000(%rbp), %rax
-1600000(%rbp), %rcx
$800000, %edx
                                                                                                                                                                        cali
                                                                                                                                                                        lead
                                                                                                                                                                        leag
                                                                                                                                                                        movl
                                                                                                                                                                                        %rcx, %rsi
%rax, %rdi
                                                                                                                                                                        movq
                                                                                                                                                                        movq
                                                                                                                                                                                        memcpy@PLT
                                                                                                                                                                        call.
                                                                                                                                                                                          -395640(%rbp).
                                                                                                                                                                                                                      %eav
```

```
c. struct as local var
```

```
const int n = 100000;
                                                                                                                                                                             Z4funci:
struct A{
                                                                                     LFB0:
                                                                                                 .cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
leal -798720(%esp), %eax
                  int arr1[n];
                                                                                                                                                                                              .cfi startproc
                                                                                                                                                                                             endbr64
                 int arr2[n];
                                                                                                                                                                                            endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq -798720(%rsp), %r11
};
int func(int k)
                  A s;
for(int i = 0; i < n; i ++) .LPSRL0:
                                                                                                                $4096, %esp

$0, (%esp)

%eax, %esp
.LPSRL0

$1296, %esp
_x86.get_pc_thunk.ax

$_GLOBAL_OFFSET_TABLE_, %eax

$0, -4(%ebp)
                                                                                                                                                                             .LPSRL0:
                                                                                                                                                                                                            $4096, %rsp
$0, (%rsp)
%r11, %rsp
.LPSRL0
$1184, %rsp
%edi, -800020(%rbp)
$0, -4(%rbp)
                                                                                                  subl
                                                                                                 orl
cmpl
                                   s.arr1[i] = i - 1;
s.arr2[i] = i;
                                                                                                                                                                                             ога
                                                                                                                                                                                            cmpq
jne
                                                                                                  jne
subl
                  return s.arr1[k];
                                                                                                                                                                                             suba
                                                                                                  call
                                                                                                                                                                                             movi
                                                                                                  addl
                                                                                                                                                                                             movl
                                                                                                  movl
                                                                                  .L3:
                                                                                                                                                                             .L3:
int main()
                                                                                                                                                                                             cmpl
                                                                                                                                                                                                             $99999, -4(%rbp)
                                                                                                                 $99999, -4(%ebp)
                                                                                                                $99999, -4(%ebp)
.L2
-4(%ebp), %eax
-1(%eax), %edx
-4(%ebp), %eax
%edx, -800004(%ebp,%eax,4)
-4(%ebp), %eax
100000(%eax), %edx
-4(%ebp), %eax
%eax, -800004(%ebp,%edx,4)
$1, -4(%ebp)
.L3
                                                                                                                                                                                             jg
movl
leal
                                                                                                  jg
movl
leal
                                                                                                                                                                                                             -4(%rbp), %eax
-1(%rax), %edx
-4(%rbp), %eax
                  A s1;
                 int n = func(348);
printf("%d",n);
                                                                                                  movl
movl
                                                                                                                                                                                            movl
cltq
                                                                                                                                                                                                             %edx, -800016(%rbp,%rax,4)
-4(%rbp), %eax
                                                                                                  movl
leal
                                                                                                                                                                                            movl
movl
                                                                                                                                                                                             cltq
leaq
                                                                                                  movl
                                                                                                                                                                                                            100000(%rax), %rdx
-4(%rbp), %eax
%eax, -800016(%rbp,%rdx,4)
$1, -4(%rbp)
.L3
                                                                                                  add1
                                                                                                                                                                                             movi
                                                                                                 jmp
                                                                                                                                                                                            movl
addl
                                                                                  .L2:
                                                                                                                8(%ebp), %eax
-800004(%ebp,%eax,4), %eax
                                                                                                                                                                                             jmp
                                                                                                  movl
leave
                                                                                                                                                                             .L2:
                                                                                                                                                                                                             -800020(%rbp), %eax
                                                                                                  .cfi_restore 5
.cfi_def_cfa 4, 4
                                                                                                                                                                                             cltq
                                                                                                                                                                                            movi
leave
                                                                                                                                                                                                             -800016(%rbp,%rax,4), %eax
                                                                                                                                                                                             .cfi_def_cfa 7, 8
                                                                                   .LFB1:
                                                                                                 .cfi_startproc
endbr32
leal 4(%esp), %ecx
.cfi_def_cfa 1, 0
andl $-16, %esp
pushl -4(%ecx)
pushl %ebp
eovl %esp. %ebp
                                                                                                                                                                             main:
                                                                                                                                                                               LFB1:
                                                                                                                                                                                                .cfi_startproc
endbr64
                                                                                                                                                                                               pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset 6, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq -798720(%rsp), %r11
                                                                                                  movl %esp, %ebp
.cfi_escape 0x10,0x5,0x2,0x75,0
                                                                                                  pushl
                                                                                                 pushl %ebx
pushl %ecx
                                                                                                   .cfi_escape 0xf,0x3,0x75,0x78,0x6
.cfi_escape 0x10,0x3,0x2,0x75,0x7c
leal -798720(%esp), %eax
                                                                                                                                                                             .LPSRL1:
                                                                                    .LPSRL1:
                                                                                                                                                                                                                   $4096, %rsp
$0, (%rsp)
%r11, %rsp
.LPSRL1
                                                                                                                                                                                                subq
                                                                                                                $4096, %esp
$0, (%esp)
%eax, %esp
.LPSRL1
                                                                                                                                                                                                ога
                                                                                                 orl
                                                                                                  cmpl
                                                                                                                                                                                                cmpq
                                                                                                  ine
                                                                                                                                                                                                jne
                                                                                                 subl
call
                                                                                                                 $1296, %esp
_x86.get_pc_thunk.bx
$_GLOBAL_OFFSET_TABLE_, %ebx
                                                                                                                                                                                                                    $1296, %rsp
$348, %edi
                                                                                                                                                                                                subq
                                                                                                                                                                                                movl
                                                                                                  addl
                                                                                                                                                                                                                      Z4funci
                                                                                                                                                                                                call
                                                                                                  pushl
                                                                                                  call
addl
                                                                                                                 _Z4funci
$4, %esp
```

d. changing Size = 500000.cfi_startproc
endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
leal -3997696(%esp), %eax .cfi_startproc
endbr64
pushq %rbp
.cfi_def_cfa_offset 16
.cfi_offset δ, -16
movq %rsp, %rbp
.cfi_def_cfa_register 6
leaq -3997696(%rsp), %r11 size(relative to c.) \$4096, %esp \$0, (%esp) %eax, %esp .LPSRL0 \$4096, %rsp \$0, (%rsp) %r11, %rsp .LPSRL0 \$2208, %rsp %edi, -400002 \$0, -4(%rbp) orl стра \$2320, %esp
_x86.get_pc_thunk.ax
\$_GLOBAL_OFFSET_TABLE_, %eax
\$0, -4(%ebp) . 0020**(%rbp)** movl movl .L3: \$499999, -4(%rbp) cmpl \$499999, -4(%ebp) jg movl leal .L2
-4(%ebp), %eax
-1(%eax), %edx
-4(%ebp), %eax
%edx, -4000004(%ebp,%eax,4)
-4(%ebp), %eax
500000(%eax), %edx
-4(%ebp), %eax
%eax, -4000004(%ebp,%edx,4)
51, -4(%ebp)
.L3 -4(%rbp), %eax -1(%rax), %edx -4(%rbp), %eax movl cltq %edx, -4000016(%rbp,%rax,4)
-4(%rbp), %eax movl movl cltq leaq 500000(%rax), %rdx -4(%rbp), %eax %eax, -4000016(%rbp,%rdx,4) \$1, -4(%rbp) .L3 movl addl jmp 8(%ebp), %eax -4000004(%ebp,%eax,4), %eax -4000020(%rbp), %eax movl .cfi_restore 5
.cfi_def_cfa 4, 4
ret cltq movl leave -4000016(%rbp,%rax,4), %eax .cfi_def_cfa 7, 8 .cfi_startproc endbr32 leal 4(%esp) main: .LFB1: 4(%esp), %ecx .cfi startproc .cfi_def_cfa 1, 0 andl \$-16, %esp pushl -4(%ecx) endbr64 pushq %rbp 1 -4(%ecx)
1 %ebp
2 %esp, %ebp
2 escape 0x10,0x5,0x2,0x75,0
1 %ebx
1 %ecx .cfi_def_cfa_offset 16 .cfi_offset 6, -16 movq %rsp, %rbp .cfi_def_cfa_register 6 -3997696(%rsp), %r11 .LPSRL1: subq \$4096, %rsp .LPSRL1: \$0, (%rsp) %r11, %rsp .LPSRL1 \$4096, %esp \$6, (%esp) %eax, %esp .LPSRL1 \$2320, %esp _x86.get_pc_thunk.bx \$_GLOBAL_OFFSET_TABLE_, ога cmpq orl cmpl jne \$2320, %rsp \$348, %edi suba movl Z4funci call

- just big numbers of allocated memory appear
- also it seems that %rsp cannot be reduced by a bigger number than 4096, thus there is a cycle .LPSRL1, which decrements %rsp only by 4096. After that %rsp is once again decremented for lacking bytes.

6. Recursion

```
64bit
                                                                    32bit
const int n = 100000;
int recursion(int a)
                                                          .cfi_startproc
                                                                                                  recursion:
                                                         endbr32
                                                                                                   .LFB0:
        int arr[n];
                                                         pushl %ebp
        arr[4] = a;
if(arr[4] < 19)
                                                         .cfi_def_cfa_offset 8
                                                         .cfi_der_c.s_
.cfi_offset 5, -8
movl %esp, %ebp
                                                                                                              .cfi_startproc
                                                                                                             endbr64
                return recursion(arr[4] + 1);
                                                                                                             pushq
                                                                                                                        %гЬр
                                                          .cfi_def_cfa_register 5
                                                         pushl
                                                                 %ebx
                                                                                                              .cfi_def_cfa_offset 16
        else return arr[4];
                                                         subl $20, %esp
.cfi_offset 3, -12
call __x86.get_pc_thunk.ax
                                                                                                              .cfi_offset 6, -16
                                                                                                             movq
                                                                                                                         %rsp, %rbp
                                                                  $_GLOBAL_OFFSET_TABLE_, %eax
int main()
                                                         addl
                                                                                                              .cfi_def_cfa_register 6
                                                         movl
                                                                  %esp, %eax
                                                                  %eax, %ebx
$100000, %eax
$1, %eax
                                                                                                                         %гЬх
                                                                                                             pushq
        int c = recursion(0);
                                                         movl
                                                                                                             subq $40, %rsp
.cfi_offset 3, -24
                                                         movl
                                                         subl
                                                                  %eax, -12(%ebp)
$100000, %eax
0(,%eax,4), %edx
                                                         movl
                                                                                                                         %edi, -36(%rbp)
%rsp, %rax
                                                                                                             movl
                                                         movl
                                                         leal
                                                                                                             mova
                                                                  $16, %eax
$1, %eax
%edx, %eax
$16, %ecx
$0, %edx
                                                         movl
                                                                                                                         %rax, %rbx
                                                                                                             movq
                                                         subl
                                                                                                                         $100000, %eax
                                                                                                             movl
                                                         addl
                                                         movl
                                                                                                             cltq
                                                         movl
                                                                                                             subq
                                                                                                                         $1, %rax
                                                         divl
                                                                  %ecx
                                                                  $16, %eax, %eax
                                                                                                                         %rax, -24(%rbp)
                                                                                                             movq
                                                         imull.
                                                                  %eax, %edx
$-4096, %edx
%esp, %ecx
%edx, %ecx
%ecx, %edx
                                                         movl
                                                                                                                         $100000, %eax
                                                                                                             movl
                                                         andl
                                                                                                             cltq
                                                         movl
                                                                                                                         %rax, %r10
                                                         subl
                                                                                                             movq
                                                         movl
                                                                                                             movl
                                                                                                                         $0, %r11d
                                                 .L2:
                                                                                                                         $100000, %eax
                                                                                                             movl
                                                         cmpl
                                                                  %edx, %esp
                                                                                                             cltq
                                                         je
subl
                                                                  .L3
                                                                  $4096, %esp
                                                                                                             movq
                                                                                                                         %rax, %r8
                                                                  $0, 4092(%esp)
.L2
                                                         orl
                                                                                                             movl
                                                                                                                         $0, %r9d
                                                         jmp
                                                 .L3:
                                                                                                                         $100000, %eax
                                                                                                             movl
                                                         movl
                                                                  %eax, %edx
                                                                                                             clta
                                                                  $4095, %edx
%edx, %esp
                                                         andl
                                                                                                             leag
                                                                                                                         0(,%rax,4), %rdx
                                                         subl
                                                                  %eax, %edx
$4095, %edx
%edx, %edx
                                                         movl
                                                                                                                         $16, %eax
                                                                                                             movl
                                                         andl
                                                                                                                         $1, %rax
                                                                                                             subq
                                                         testl
                                                         je
andl
                                                                                                             addq
                                                                                                                         %rdx, %rax
                                                                  $4095, %eax
                                                                                                                         $16, %esi
                                                                                                             movl
                                                        subl
                                                                  $4, %eax
                                                                                                             movl
                                                                                                                         $0, %edx
                                                                  %esp, %eax
$0, (%eax)
                                                         addl
                                                                                                             divq
                                                                                                                         %rsi
                                                         orl
                                                                                                                         $16, %rax, %rax
                                                                                                             imulq
                                                                                                                         %rax, %rdx
                                                                                                             mova
                                                                                                                         $-4096, %rdx
                                                                                                             andq
                                                                                                                         %rsp, %rcx
                                                                                                             movq
                                                                                                                         %rdx, %rcx
                                                                                                             subq
                                                                                                                         %rcx, %rdx
                                                                                                             movq
                                                                                                  .L2:
                                                                                                                         %rdx, %rsp
                                                                                                             cmpq
                                                                                                             jе
                                                                                                                         .L3
                                                                                                                         $4096, %rsp
                                                                                                             subq
                                                                                                                         $0, 4088(%rsp)
                                                                                                             ога
                                                                                                             jmp
```

```
.L4:
                                      L3:
         movl
                  %esp, %eax
                                                        %rax, %rdx
                                               movq
         addl
                  $3, %eax
                                               andl
                                                         $4095, %edx
         shrl
                  $2, %eax
                                               subq
                                                        %rdx, %rsp
                  $2, %eax
         sall
                                                        %rax, %rdx
                                               movq
                  %eax, -16(%ebp)
         movl
                                               andl
                                                         $4095, %edx
                  -16(%ebp), %eax
         movl
                                               testq
                                                        %rdx, %rdx
         movl
                  8(%ebp), %edx
                                               je
                                                         .L4
                  %edx, 16(%eax)
         movl
                                               andl
                                                         $4095, %eax
         movl
                  -16(%ebp), %eax
                                                        $8, %rax
                                               subq
                  16(%eax), %eax
         movl
                                               addq
                                                        %гѕр, %гах
                  $18, %eax
         cmpl
                                                         $0, (%rax)
                                               ога
                  .L5
         jg
                                      L4:
                  -16(%ebp), %eax
         movl
                                                        %rsp, %rax
                                               movq
         movl
                  16(%eax), %eax
                                               addq
                                                        $3, %rax
         addl
                  $1, %eax
                                                        $2, %rax
                                               shrq
         subl
                  $12, %esp
         pushl
                                               salq
                                                        $2, %rax
                  %eax
         call
                  recursion
                                                        %rax, -32(%rbp)
                                               movq
                                                         -32(%rbp), %rax
         addl
                  $16, %esp
                                               movq
                  .L6
         jmp
                                               movl
                                                         -36(%rbp), %edx
.L5:
                                               movl
                                                        %edx, 16(%rax)
         movl
                  -16(%ebp), %eax
                                                         -32(%rbp), %rax
                                               movq
         movl
                  16(%eax), %eax
                                                         16(%rax), %eax
                                               movl
.L6:
                                               cmpl
                                                        $18, %eax
         movl
                  %ebx, %esp
                                                         .L5
                                               jg
                  -4(%ebp), %ebx
         movl
                                                         -32(%rbp), %rax
                                               movq
         leave
                                               movl
                                                         16(%rax), %eax
         .cfi_restore 5
                                               addl
                                                        $1, %eax
         .cfi_restore 3
                                                        %eax, %edi
                                               movl
         .cfi_def_cfa 4, 4
                                               call
                                                         recursion
         ret
                                               jmp
                                                         .L6
main:
                                      L5:
.LFB1:
      .cfi_startproc
                                                         -32(%rbp), %rax
                                               pvom
      endbr32
                                               movl
                                                         16(%rax), %eax
            4(%esp), %ecx
      leal
      .cfi_def_cfa 1, 0
andl $-16, %esp
                                      L6:
      andl
                                                        %rbx, %rsp
                                               pvom
             -4(%ecx)
      pushl
                                               pvom
                                                         -8(%rbp), %rbx
             %ebp
      pushl
      movl
             %esp, %ebp
                                               leave
      .cfi_escape 0x10,0x5,0x2,0x75,0
                                               .cfi_def_cfa 7, 8
      pushl
            %ecx
       .cfi_escape 0xf,0x3,0x75,0x7c,0x6
      sub1
             $20, %esp
             __x86.get_pc_thunk.ax
$_GLOBAL_OFFSET_TABLE_, %eax
                                      main:
      call
      addl
                                       .LFB1:
      subl
             $12, %esp
                                                .cfi_startproc
      pushl
             S0
      call
             recursion
                                               endbr64
             $16, %esp
      addl
             %eax, -12(%ebp)
$0, %eax
                                               pushq
                                                         %гьр
      movl
                                                .cfi_def_cfa_offset 16
      movl
             -4(%ebp), %ecx
      movl
                                                .cfi_offset 6, -16
                                               movq
                                                         %rsp, %rbp
                                                .cfi_def_cfa_register 6
                                                         $16, %rsp
                                               subq
                                               movl
                                                         $0, %edi
                                               call
                                                         recursion
                                               movl
                                                         %eax, -4(%rbp)
                                               movl
                                                         $0, %eax
                                               leave
                                                .cfi_def_cfa 7, 8
                                               ret
```

```
C++
                                                                     32bit
                                                                                                                       64bit
#include <iostream>
                                                  _Z9recursioni:
                                                                                                    _Z9recursioni:
                                                  .LFB1519:
                                                                                                     .LFB1522:
const int n = 100000;
                                                           .cfi_startproc
                                                                                                               .cfi_startproc
                                                          endbr32
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
int recursion(int a)
                                                                                                               endbr64
                                                                                                               pushq
                                                                                                                         %гЬр
        int arr[n];
                                                                                                               .cfi_def_cfa_offset 16
        arr[4] = a;
if(arr[4] < 12)
                                                                                                               .cfi_offset 6, -16
                                                           .cfi_def_cfa_register 5
leal -397312(%esp), %eax
                                                                                                                        %гѕр, %гЬр
                                                          leal
                                                                                                               movq
                 return recursion(arr[4] + 1); .LPSRL0:
                                                                                                               .cfi_def_cfa_register 6
                                                                   $4096, %esp
                                                          subl
                                                                                                                          -397312(%rsp), %r11
                                                                                                               leag
        else return arr[4];
                                                                   $0, (%esp)
                                                          orl
                                                                                                     .LPSRL0:
                                                                   %eax, %esp
.LPSRL0
                                                          cmpl
                                                          jne
                                                                                                               subq
                                                                                                                          $4096, %rsp
int main()
                                                           subl
                                                                   $2712, %esp
                                                                                                               ога
                                                                                                                          $0, (%rsp)
                                                                   __x86.get_pc_thunk.ax
$_GLOBAL_OFFSET_TABLE_, %eax
                                                          call
                                                                                                               cmpq
                                                                                                                          %r11, %rsp
        int c = recursion(0);
                                                          addl
                                                                                                                          .LPSRL0
                                                                   %gs:20, %eax
%eax, -12(%ebp)
%eax, %eax
                                                                                                               jne
                                                          movl
                                                                                                                          $2704, %rsp
                                                          movl
                                                                                                               subq
                                                          xorl
                                                                                                               movl
                                                                                                                          %edi, -400004(%rbp)
                                                                   8(%ebp), %eax
%eax, -399996(%ebp)
-399996(%ebp), %eax
                                                          movl
                                                                                                                          -400004(%rbp), %eax
%eax, -399984(%rbp)
                                                                                                               movl
                                                          movl
                                                                                                               movl
                                                          movl
                                                                                                                          -399984(%rbp), %eax
                                                                   $18, %eax
                                                                                                               movl
                                                          cmpl
                                                          jg
movl
                                                                                                               cmpl
                                                                                                                          $18, %eax
                                                                   -399996(%ebp), %eax
                                                                                                               jg
                                                                                                                          .L2
                                                          addl
                                                                   $1, %eax
                                                                                                                          -399984(%rbp), %eax
                                                                                                               movl
                                                          subl
                                                                   $12, %esp
                                                          pushl
                                                                   %eax
                                                                                                               addl
                                                                                                                          $1, %eax
                                                                    Z9recursioni
                                                          call
                                                                                                               movl
                                                                                                                          %eax, %edi
                                                                   $16, %esp
                                                          addl
                                                                                                                          _Z9recursioni
                                                                                                               call
                                                          jmp
                                                                   .L4
                                                                                                                          .L4
                                                                                                               jmp
                                                  .L2:
                                                          movl
                                                                   -399996(%ebp), %eax
                                                                                                     .L2:
                                                                                                                          -399984(%rbp), %eax
                                                                                                               movl
                                                          movl
                                                                   -12(%ebp), %edx
                                                                                                     .L4:
                                                                   %gs:20, %edx
                                                          xorl
                                                          jе
                                                                                                               leave
                                                          call
                                                                   __stack_chk_fail_local
                                                                                                    main:
                                                          leave
                                                                                                     .LFB1523:
                                                  main:
                                                                                                                .cfi_startproc
                                                  .LFB1520:
                                                                                                                endbr64
                                                          .cfi_startproc
                                                                                                                pushq
                                                                                                                            %гьр
                                                          endbr32
                                                                   4(%esp), %ecx
                                                          leal
                                                                                                                .cfi def cfa offset 16
                                                          .cfi_def_cfa 1, 0
andl $-16, %esp
                                                                                                                 .cfi_offset 6, -16
                                                                   -4(%ecx)
                                                          pushl
                                                                                                                pvom
                                                                                                                            %rsp, %rbp
                                                                   %ebp
                                                          pushl
                                                                                                                .cfi_def_cfa_register 6
                                                          movl
                                                                   %esp, %ebp
                                                                escape 0x10,0x5,0x2,0x75,0
                                                                                                                subq
                                                                                                                            $16, %rsp
                                                           .cfi
                                                          pushl
                                                                   %ecx
                                                                                                                movl
                                                                                                                            $0, %edi
                                                          .cfi_escape 0xf,0x3,0x75,0x7c,0x6
                                                                                                                call
                                                                                                                             Z9recursioni
                                                                   $20, %esp
__x86.get_pc_thunk.ax
$_GLOBAL_OFFSET_TABLE_, %eax
                                                          subl
                                                          call
                                                                                                                            %eax, -4(%rbp)
                                                                                                                movl
                                                          addl
                                                                                                                movl
                                                                                                                            $0, %eax
                                                          subl
                                                                   $12, %esp
                                                                   $0
                                                                                                                leave
                                                          pushl
                                                                   _Z9recursioni
$16, %esp
                                                          call
                                                                                                                .cfi_def_cfa 7, 8
                                                          addl
                                                                   %eax, -12(%ebp)
$0, %eax
-4(%ebp), %ecx
                                                                                                                ret
                                                          movl
                                                          movl
                                                          movl
                                                           .cfi_def_cfa 1, 0
                                                          leave
                                                           .cfi_restore 5
                                                          leal -4(%ecx), %esp
.cfi_def_cfa 4, 4
```

• Differences between 32bit and 64bit are not that significant, just the amount of memory allocated for arrays(differs by approximately 0-20 bytes)

```
#include <iostream>
const int n = 100000;
int recursion(int a)
{
    int arr[n];
    arr[4] = a;
    if(arr[4] < 10)
    {
        return recursion(arr[4] + 1);
    }
    else return arr[4];
}
int main()
{
    int c = recursion(0);
}</pre>
```

program executes correctly, when arr[4] < 19, so for 20 recursion calls. If I write arr[4] < 20 – stack overflow occurs. So the estimated stack size is between 20*100000*4 bytes \approx 7800KB and 8200KB. So assuming that stack size is a good looking number, it's around 8MB.