Secure and Dependable Systems
Arsenij Percov

Jacobs University Bremen

March 22, 2020

Problem Sheet 4

Problem 4.1

Solution:

```
Dump of assembler code for function foo:
0x0000555555555125 <+0>: push %rbp \\push the address of the base pointer to the stack to
0x0000555555555126 <+1>: mov %rsp,%rbp \\ make rbp point to the top of the stack
0x0000555555555129 <+4>: mov %edi,-0x14(%rbp)\\store %edi at address that is 20 bytes less
0x000055555555512c <+7>: movl $0x0,-0x8(%rbp)\\ store 0 at rbp - 8 (allocating of local va
0x0000555555555133 <+14>: movl $0x1,-0x4(%rbp)\\ store 1 at rbp - 4 (allocating of local v
0x000055555555513a <+21>: jmp 0x555555555146 <foo+33>\\jump to address provided as argumen
0x000055555555513c <+23>: mov -0x4(%rbp),%eax \\ load value of rbp - 4 to eax (in our case
0x000055555555513f <+26>: add %eax,-0x8(%rbp)\\ add value store in eax (i) to value at add
0x0000555555555142 <+29>: addl $0x1,-0x4(%rbp)\\add 1 to value stored in rbp-4 (i)
0x0000555555555146 <+33>: mov -0x4(%rbp),%eax \\ load value of rbp  - 4(i) to eax
0x0000555555555149 <+36>: cmp -0x14(%rbp),%eax \\ compare whether rbp - 20 (a) is equal to
0x000055555555514c <+39>: jle 0x55555555513c <foo+23>\\ jump to the address, jump to the b
0x000055555555514e <+41>: mov -0x8(%rbp),%eax \\move value store rbp - 8 (r) to eax , savi
0x0000555555555151 <+44>: pop %rbp \\pop base pointer from stack
0x0000555555555152 <+45>: retq\\return to the caller
```