

# ICS Homework 9

November 26, 2021

## 1. Procedure Call

Tower of Hanoi is a mathematical puzzle where we have three rods and  $n$  disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

- 1) Only one disk can be moved at a time.
- 2) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
- 3) No disk may be placed on top of a smaller disk.

To solve this problem, we have the following hanoi function, which moves  $n$  disk from  $src$  to  $dst$  by using  $helper$ . Notice that the type of index (of rods) is `char`.

```
void hanoi(int n, char src, char helper, char dst){
    if(n==1)
        move(src,dst);
    else {
        hanoi(n-1, src, dst, helper);
        move(src,dst);
        hanoi(n-1, helper, src, dst);
    }
}
```

And we also have the function `move`, which executes the move action to move the upper disk from  $src$  to  $dst$ .

```
void move(char src, char dst);
```

The assembly code of the `hanoi` function is shown below. Answer the following questions.

1. hanoi:	25. movsbl <u>(4)</u> , %ecx <i>(4) -12(2rbp)</i>
2. pushq %rbp	26. movsbl <u>(5)</u> , %eax <i>(5) -8(2rbp)</i>
3. movq %rsp, %rbp	27. movl <u>(6)</u> , %esi <i>(6) -4(2rbp)</i>
4. subq \$16, %rsp	28. leal <u>(7)</u> , %edi <i>(7) -1(2rbp)</i>
5. movl %esi, %eax <i>src</i>	29. movl %eax, %esi <i>(7) -1(2rbp)</i>
6. movl %edi, -4(%rbp) <i>n</i>	30. call hanoi
7. movl %edx, %esi <i>helper</i>	31.
8. movl %ecx, %edx <i>dst</i>	32. movsbl <u>(8)</u> , %edi <i>-8(2rbp)</i>
9.	33. movsbl <u>(9)</u> , %esi <i>-16(2rbp)</i>
10. movb %al, -8(%rbp) <i>src</i>	34. call move
11. movl %esi, %eax	35.
12. movb %al, -12(%rbp) <i>helper</i>	36. movsbl <u>(10)</u> , %eax <i>(10) -12(2rbp)</i>
13. movl %edx, %eax	37. movsbl <u>(11)</u> , %edx <i>(11) -8(2rbp)</i>
14. movb %al, -16(%rbp) <i>dst</i>	38. movl <u>(12)</u> , %esi <i>(12) -4(2rbp)</i>
15. cmpl \$1, -4(%rbp)	39. leal <u>(13)</u> , %edi <i>(13) -1(2rbp)</i>

16.   jne     .branch1	40.   movl   %eax, %esi
17.       -16(8rbp)	41.   movsbl (14), %ecx
18.   movsbl (1), %esi	42.   call   hanoi -16(8rbp)
19.   movsbl (2), %edi	43.
20.   call   move -8(8rbp)	44. .finish:
21.   jmp     .finish	45.   nop
22.	46.   leave
23. .branch1: -16(8rbp)	47.   ret
24.   movsbl (3), %edx	

1. Find out where the local variables stored in the stack and fill the following table using %rbp.

n	src	helper	dst
-4(8rbp)	-8(8rbp)	-12(8rbp)	-16(8rbp)

2. Fill the blanks in the assembly code.