CSCI 361 Week 13 Notes

Kyle Krstulich

April 17, 2025

Announcements

Chapter 8 will be the last major programming assignment. May 1st will be exam 2, while the final is on the syllabus, Tuesday on finals week.
Helper functions due Thursday, while next week we will cover the harder sections.

Function Calls

High level language:

```
if (cond)
  s1
else
  s2
```

In order to translate this to a VM language. (not pythonic indentation)

```
~cond (n = not)
if-goto L1
  s1
  goto L2
label L1
  s2
label L2
```

To call a function you must first push all arguments to stack.

Call Stack and Recursion

```
subroutine a:
  call b
  call c
subroutine b:
  call c
  call d
subroutine c:
  call d
subroutine d:
  null
```

Trace:

```
call a
  call b
  push a
    call c
    push b
      call d
      push c
      return d
    return c
    pop c
  call d
    push b
    return d
  pop b
call c
  push a
  call d
    push c
    pop c
  pop a
pop a
```

Function POV

Calling function POV

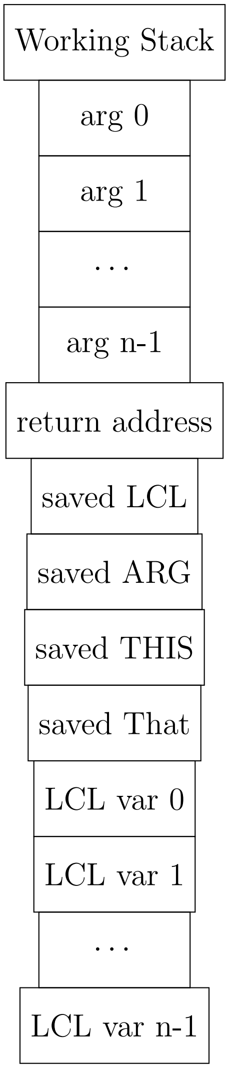
- push arguments to stack
- Static segment is set based on filename.
- 'call' function (jump to that code)
- return value. Top of stack contains the function return value.
- memory segments; argument, local, static, this, that, pointer. Are as they were before the call. Function call does not mutate the memory segments.

POV of called function

- Local segment initialized/reserved set to 0 (in hack).
- Static segment is set based on filename.
- Working stack appears to be empty. this, that, pointer, temp are undefined. This defines a pristine workspace for the function.
- will push return values to top of stack.

Function Call Implementation

At the point of calling a function:



- **Return Address**: uses push label. Lets us know where to come back to after function call is finished. Is a ROM address.

- **Frame**: the LCL, ARG, THIS, THAT, and return address.
- **Local Variables**: LCL var n
- **LCL, ARG, THIS, THAT** uses pushMem()

Overall handing a function call:

- Save the return address
- Save the callers segment pointers (LCL, ARG, THIS, THAT)
- Reposition ARG (for the callee) argmuments are located after the caller working stack.
- Reposition LCL (for the callee)
- Go to execute the callee's code (@function, 0;JMP)