10/19/16 Assembly

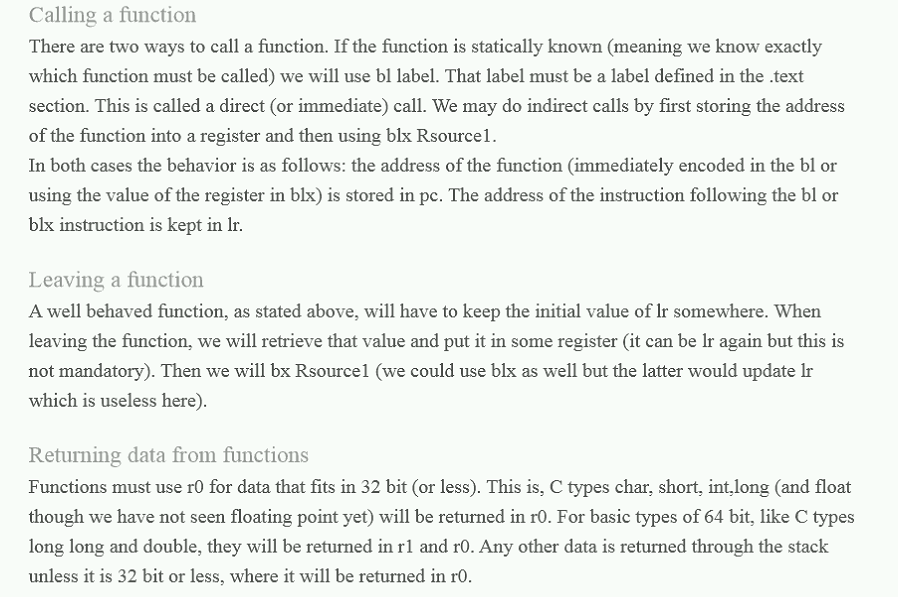
Class Notes

ABI

* Application binary interface
* Standard for assembly
  + AAPCS
    - Another standard
    - Arm architecture procedure call standard
* RV – return value
* Standard:
  + R0 – first param/return value\*
    - From 64bit to lower 32 bit
  + R1 – second param/return value
    - For 64 bit data types
  + R2 – third param
  + R3 – fourth param
  + Additional param pushed onto stack

Passing values

* In c++, two types:
  + Pass-by-value
  + Pass-by-reference
    - Pointer
* In asm, two ways
  + bl <label>
    - label must be deinfed in .text
  + blx Rsource1
    - Direct/immediate call
    - Alt: bx Rsource1

Function summary : 

LSL

* Def.) Logic shift left
* Double the number
* Ex) Calculating 15\*10 via shifts

Mov r0, #15

Mov r1, r0

Mov r0, r0, LSL #1

Mov r1, r1, lsl #3

Add r0,r0,r1

* See PDF for details
* ex)



Recurrsion

* A function that calls itself
  + Most recursion functions can be replaced with a while loop
* Ex) Ackermann’s function
* Parts of a recursive function:
  + 1. Exit condition (loop terminator)
  + 2. Recursive call
* Factorial Ex)
  + n! = n+ (n-1) +(n-2) …1
    - 0! = 1
    - 1! = 1
    - 2! = 2+1 = 3
    - 3! = 3+2+1 = 6
    - Etc.
  + Shortcut
    - 5! = 5\*4!
    - 6! = 6\*5!
* C++ Ex)
  + Int fact(int n){

If(n== 0)

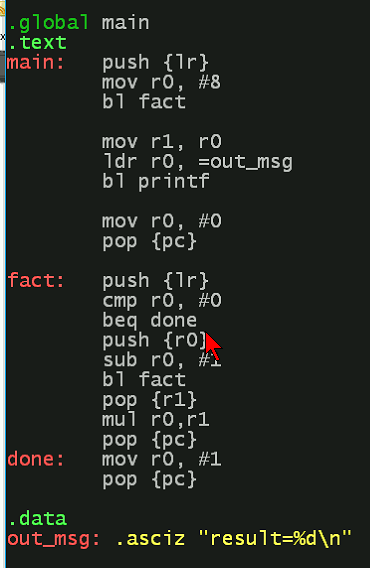
return 1; //Exit condition

else

return n\*fact(n-1); //Shortcut method

}

* + Once the function reaches 0, the function “unwinds” itself and begins returning the answer to solve the factorial.
* Asm Ex)



* + Address for {lr} changes based on where it is called.
    - Ex) first *push{lr}* for fact: is contained in *main* where *bl fact* is
      * Second *push {lr}* is contained in *fact* where *bl fact* is