CS2208 Assignment 5

Nicholas Tocco
SID: 250909544
April 03, 2018

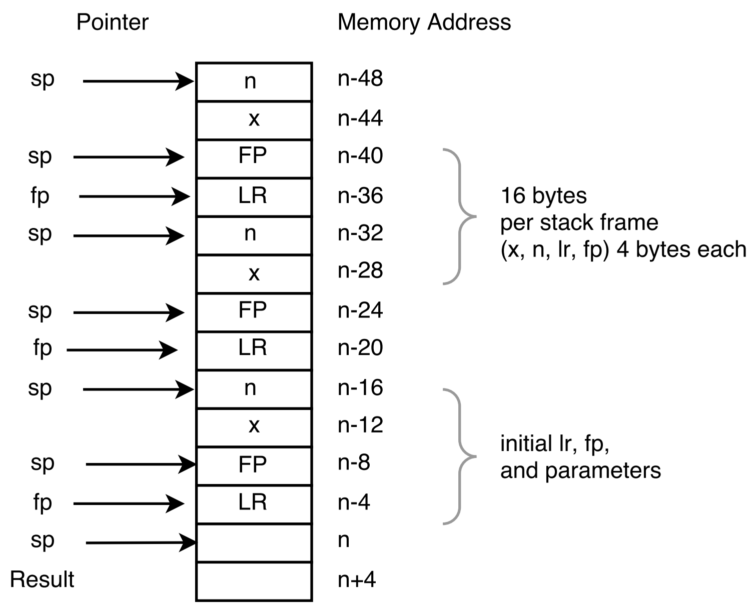
Source Code

```
AREA recursion1, CODE, READONLY
            ENTRY

x           equ 2               ;directive for x
n           equ 4               ;directive for n

main        adr sp, STACK       ;stack pointer points to allocated space referenced by STACK
            mov r0, #x          ;put value in r0 (x)
            mov r1, #n          ;put value in r1 (n)
            bl power            ;brach to recursive power function
            adr r3, RESULT      ;get address of RESULT
            str r1, [r3]        ;store return value of POWER in RESULT
exit        b exit              ;FINISHED!

power       stmdb r13!, {fp, lr}  ;store the fp (initially 0) and the lr onto the stack
            add fp, sp, #4      ;set the fp to the sp+4, at the base of the current stack frame
            sub sp, sp, #8      ;advance stack pointer to the top of the stack, reserving 8 bytes for local variables
            str r0, [fp, #-8]   ;store x on the stack 8 bytes away from the fp
            str r1, [fp, #-12]  ;store n on the stack 12 bytes away from the fp
            ldr r0, [fp, #-8]   ;load current x into r0
            ldr r1, [fp, #-12]  ;load current n into r1
            cmp r1, #0          ;check if n==0
            bne notZero         ;if n!=0 , branch to notZero
            mov r1, #1          ;if n==0, set return value n = 1
            b return            ;and branch to return

notZero     tst r1, #1          ;test last bit in n to determine if even or odd
            beq even            ;branch to even if last bit is not set, continue if it is set
            sub r1, r1, #1      ;subtract 1 from n
            bl power            ;recursive call power(x, n-1) (r0,r1)
            ldr r1, [fp, #-8]   ;load local variable x into r1 (stored 8 bytes away from fp)
            mul r1, r0, r1      ;multiply x (r1) by the result of (x,n-1)  (r0)
            b return            ;return

even        lsr r1, r1, #1      ;shift n to the right by 1 bit
            bl power            ;recursive call power(x, n>>1) (r0,r1)
            mov r1, r0          ;r0 hold the current y value -- move into r1 to multiply
            mul r1, r0, r1      ;multiply y*y and store result in r1

return      mov r0, r1          ;move result (return value) into r0
            sub sp, fp, #4      ;set stack pointer to 4 bytes below fp (on previous lr and fp)
            ldmia r13!, {fp, lr}  ;load the previous frame pointer and lr
            bx lr               ;branch to last call

            SPACE 512           ;lots of stack space just to be safe
STACK       DCD 0x00            ;location of stack in memory (FD)
RESULT      DCD 0x00;           ;to store the result of x^n
            END
```

Structure of the Stack Frame

Stack frame with 2 recursive calls made:

(Full decrementing stack)


Pointer
Memory Address
sp
n
n-48
x
n-44
sp
FP
n-40
fp
LR
n-36
sp
n
n-32
x
n-28
sp
FP
n-24
fp
LR
n-20
sp
n
n-16
x
n-12
sp
FP
n-8
fp
LR
n-4
sp
n
Result
n+4
16 bytes per stack frame (x, n, lr, fp) 4 bytes each
initial lr, fp, and parameters