MC48 Lab 2:
More Advanced Assembly Language Programming

Max Hailperin

Due: November 2, 1995

The goal of this lab is to practice more advanced assembly language programming skills, in particular, the use of stack frames to support recursion.

In particular, you will write an assembly language program that generates a tree-recursive process for solving a classic mathematical recreation, the "Towers of Hanoi" puzzle. This puzzle has three pegs, numbered as pegs 1, 2, and 3, and n disks, labeled as disk 1 through disk n. [The number n can vary; typical values might be in the range from 1 to 8.] Disk 1 is smaller than disk 2, which is in turn smaller than disk 3, and so forth, with disk n being the largest. Initially all the disks are on peg 1, starting with disk n on the bottom, disk $n-1$ on top of that, and so forth, up to disk 1 on the top. The goal is to move all the disks to peg 2. You may only move one disk at a time: the top disk from any of the three pegs onto the top of either of the other two peg. Moreover, there is a constraint: you must not place a larger disk on top of a smaller.

Your lab write up can be short and sweet, but it should be English. You should communicate everything you did at some level of detail, but for the uninteresting parts, you can summarize in broad outline, while for the more interesting parts, you can go into detail. Specific details I will expect to see are the program you are asked to write and test data and results from that program.

You can again use either NeXTspim or Xspim, as in lab 1.

The following is a Scheme version of the program you need to write; it prompts for and reads in n and then prints out a list of the necessary disk movements:

```
(define main
  (lambda ()
    (display "Enter number of disks> ")
    (let ((n (read)))
      (hanoi n 1 2 3))))

(define hanoi ;move n smallest disks from start to finish using extra
  (lambda (n start finish extra)
    (if (= n 0)
        'done
        (begin
          (hanoi (- n 1) start extra finish)
          (display "Move disk ")
          (display n)
          (display " from peg ")
          (display start)
          (display " to peg ")
          (display finish)
          (display ".")
          (newline)
          (hanoi (- n 1) extra finish start)))))
```

To get the strings displayed, see page A-46; the newline at the end of each line can be accommodated by using a `\n` (newline) character at the end of the last string, i.e.:

```
        .asciiz ".\n"
```

would be the string to print out last, containing both the period and the newline.