

CPSC 321:501-503 - Computer Architecture
Texas A&M University
Department of Computer Science

Fall 2006

Lab 4 (100 pts) – Towers of Hanoi
Complete by yourself

Release date: 25 September 2006

Due date: One week after the lab

1. Towers of Hanoi problem

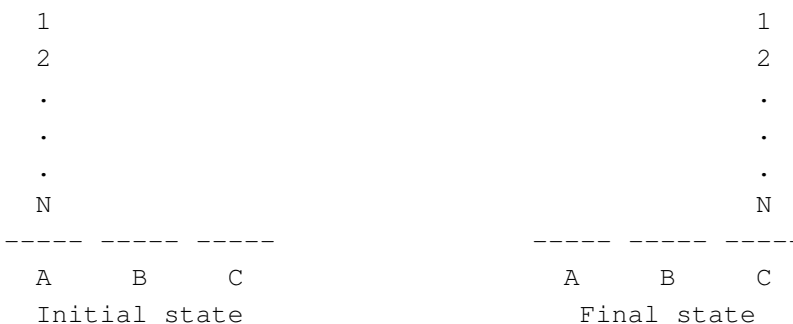
Facts (from www.lawrencehallofscience.org):

The Tower of Hanoi (sometimes referred to as the Tower of Brahma or the End of the World Puzzle) was invented by the French mathematician, Edouard Lucas, in 1883. He was inspired by a legend that tells of a Hindu temple where the pyramid puzzle might have been used for the mental discipline of young priests. Legend says that at the beginning of time the priests in the temple were given a stack of 64 gold disks, each one a little smaller than the one beneath it. Their assignment was to transfer the 64 disks from one of the three poles to another, with one important provision; a large disk could never be placed on top of a smaller one. The priests worked very efficiently, day and night. When they finished their work, the myth said, the temple would crumble into dust and the world would vanish.

The objective of this assignment is to write a MIPS assembly program to solve the *Towers of Hanoi* problem and to demonstrate the steps from the initial to the final configuration. Accept as input, the number of disks (N) to be moved (*Assume the number will be ≤ 32*).

Problem formulation

There are N disks labeled 1..N, with the larger label indicating a larger disk. These disks can be stacked on poles A, B and C. Initially all disks form a tower on pole A, as shown.



The problem is to move the tower from A to C with the help of tower B, according to the following rules:

- (a) Only one disk can be moved at a time;
- (b) No disk can rest on a smaller disk.

If we write `hanoi(N, source, dest, aux)` to denote “pass N disks from pole source to pole dest using pole aux”, then the problem is:

```
hanoi(N, A, C, B)
```

Rule (b) implies that for disk N (the largest disk) to be moved to C, the N-1 smaller disks on top of it must first be placed on B, so that disk N can then pass from A to C. To finish the solution then requires passing the N-1 disks from B to C. This implies a recursive strategy that can be written as follows:

```
hanoi ( N, source, dest, aux )
{
  if N > 0 then
    hanoi(N - 1, source, aux, dest)
    <move disk N from source to dest>
    hanoi(N - 1, aux, dest, source)
  end if
}
```

Along with solving the problem, you need to illustrate the state of the three towers after every disk move. So for N=3, you would start by showing the initial configuration of the towers, and then each step:

```
=====

      *
     **
    ***
-----
   A       B       C
-----

=====

      **
     ***
-----
   A       B       C
-----

=====

     ***      **      *
-----
   A       B       C
-----

=====
```

```

          *
        ***
-----
      A      B      C
=====

```

```

          *
         **
        ***
-----
      A      B      C
=====

```

...

(Skipping a few steps)

...

And finally:

```

                    *
                   **
                  ***
-----
      A      B      C
=====

```

The tasks for this lab are:

1. Write a recursive MIPS assembly program to solve the *Towers of Hanoi* problem.
2. Write a MIPS function `printTowers()` to display the contents of the three towers at any given instance.