Programming Assignment # 3

Code Bumming!
Session 2

1. Objectives

The tower of Hanoi problem has been well known to computer engineers for the elegance of its solution. (If you do not know the tower of Hanoi problem, please read “Tower of Hanoi” in Wikipedia first.) In this session, you are supposed to write assembly code for the tower of Hanoi solver. The solver should be implemented as a function named “hanoi” that returns an array of disc moves. The resulting code must be efficient and made with as few instructions as possible. Your code must be compiled successfully to an object file (*.o) and must work correctly when they are called from C source files.

2. Details

A. There are three pegs in this problem, and the pegs are numbered 1, 2 and 3 respectively. The pegs are represented as integer numbers in your function.

B. You are given a skeleton assembly file for the “Hanoi” function and the C file to check the functionality of your implementation. These files can be compiled into the executable file by invoking ‘make’ command in the directory.

C. The data structure for a disc move is defined as follows:

```c
typedef struct move {
    int source;
    int destination;
} move;
```

`source` refers to the source peg, and `destination` means the destination peg.

D. The prototype of the function, Hanoi, is as follows:

```c
int hanoi (move *move_list, int *num_move, int num_disc, int frompeg, int auxpeg, int topeg);
```

`move_list` points to an array of moves, of which size is MAXMOVE. By default, MAXMOVE is 10,000. The function sets `num_move` to be the number of moves in the solution. `num_disc` is the number of discs on the source peg to move. `frompeg`, `auxpeg` and `topeg` are the indexes of the source peg, auxiliary peg and destination peg, respectively. This function returns -1 when it finds out that the `num_move` exceeds 10,000. Otherwise, it returns 0.

E. These files are for IA32 Linux systems. If you have installed an x86-64 Linux distribution, install gcc-multilib support to your system. It will enable your system to compile IA32 assembly codes.

F. Code with fewer instructions will earn better score.

G. Any reverse engineering (from C code) is strictly prohibited. Think and write only in assembly.

H. The skeleton assembly files can be downloaded at
I. Do not modify any part of the C file.

3. Logistics

A. The completed source files should be tar-and-gzipped into a file. The name of the zipped file should be "studentid.tgz" (e.g. 2012310123.tgz)

C. Prepare a separate document in PDF format, which explains the design and implementation of your code. The document file name should be "studentid.pdf" (e.g. 2012310123.pdf)

D. Send a mail to [homework.skku@gmail.com] with attaching the two files, the zipped source code file and documentation. The subject of the mail should be [SWE2001 PA#3] studentID.

E. Only the assignments submitted before the deadline will receive the full credit. 25% of the credit will be deducted for every single day delay.