COSC4503 PROGRAM TRANSLATION

Program Assignment #0 (LHF)

From C-- to MIPS In 1 Week

This purpose of this warm-up assignment is introduce you to the world of program translation in one week. This assignment involves the translation of a C-- source code file into a MIPS assembly file.

The source code file is a program in the C-- language. The source code is g++ compile-able, so the output is clearly understood and there is no need for additional error-handling in this assignment.

The output file is a complete MIPS assembly code file that will run on the MARS MIPS-simulator.

For example, given the following .cmm source file

```
#include <iostream>
using namespace std;

// copy any comments from the source code

int main(int argc, char** argv) {

      int s = 123;
      s = 456;
      cout << s;

      return 0;
}
```

Something similar to the following MIPS translation would be generated by your program

```
#   copy any comments from the source code
      .data
#     int s = 123;
s:    .word     123
      .text
#     s = 456;
      la   $t0, s
      li   $t8, 456
      sw   $t8, 0($t0)
#     cout << s;
      lw   $a0, 0($t0)
      li   $v0, 1
      syscall
#    return 0
      li   $v0, 10           # system call for exit
      syscall                # program terminated.
```

INPUT

1) Allow the user to enter the C-- source code filename as a command line argument.

OUTPUT

2) The output file will share the filename of the source code with the addition of the ".asm" extension.

NOTE

3) Standalone comment lines will be copied from the source code to the assembly file.
4) The Cmm statement appears as a comment in the MIPS code, preceding the assembly translation of the statement.
5) Assume that each program concludes with a "`return 0;`" statement.
6) Sequence is the only control structure permitted in the C-- programs.
7) The only statements are assignment and output to console.
8) The assignment statement is limited to a single integer literal on the right hand side.
9) The output statement is limited to a single integer variable argument.
10) There is only one function defined in the C-- program, the **main()**.
11) All C-- variable are of type **int** and each variable is declared on a separate line as the first lines in the main() function.
12) There are at most 8 C-- variables and each has a unique single character name in the range [a..z].
13) The C-- variables are translated into the MIPS temporary registers $t0-$t7. Temporary register $t8 is used in the assignment statement translation.

(list additional notes given in class below)