

CMSC430 Code generator – Version 1
Project 4 – Due 11:59pm May 3, 2007

1. Code generator

Complete your NIP compiler.

1. Generate code for the HAC 430 computer. The format for the output from your NIP compiler is specified in section 2 - Loader Format in version 2007-2 of the NIP documentation given out earlier.
2. The basic design of your code generator is as follows:
 - (a) The basic design will be a standard two pass structure. The file created by Parser will be reread, and code will be generated on this second pass.
 - (b) By default, the output from your compiler will be the file nip.out.
 - (c) For each identifier, find the name in the symbol table. Search the symbol table by scope to get the appropriate declaration. Functions that are forward references can cause problems. Problems can occur even in innocently looking statements:
 $I := J$
where J is a globally defined INTEGER and also a forward referenced-proc.
 - (d) Any IDs resolving to level 0 symbol table now refer to undeclared identifiers.
3. To invoke compiler, user types `nip nip-file` or `nip -options nip-file`. If you use format: `nip < nip-file`, then write a small shell script wrapper that takes required format and maps it into your needed format.
4. Major requirement is to generate correct code so that program always executes correctly.
 - Minor requirement is to generate good code that executes efficiently.
 - It is worth more to get a compiler to work correctly than to have one that executes more efficiently, but sometimes generates incorrect code.