Ansifront beta release v0.12

This is a beta release version of Ansifront. Only the binary is provided. The program takes a pre-processed ANSI C program from stdin and writes a MW C program to stdout. This means you will need to run c.prep on your files first, then run Ansifront on the pre-processed file. Then compile the resulting file. Version 2.5.0 of CC contains an option to include ansifront when compiling which makes this alot easier.

Ansifront uses the "memcpy" in the code it outputs to copy structs and unions, so you will need this function in your library. Also note that the message "#####/##### stack" will be written to stderr after ansifront completes. This indicates how much memory is being used to process the file. If the number goes over 9000 or so, the file is coming close to being too big to handle. Let me know if this limitation causes problems.

Ansifront generates a file that has the same name as the source file except with the .ntbl extension (stands for name table). This file lists all the hashed names that might be unresolved at link time along with the original name. If, during linking, you have an unresolved name that is of the form _xxxxxx where 'x' is a hex digit, you can look in the .ntbl file to find out what the actual name was.

Current features:

* prototype style function signatures.
* struct/union passing to/from functions.
* struct/union assignment.
* all basic types except unsigned long, including void and void *.
  Any use of an unsigned long will be treated as a long.
* initialization of auto structs and arrays. Initialization of unions
  is explicitly disallowed because of a bug in the MW C.
* handling of const.
* enumerations
* no conflict between field names of different struct/unions or variable
  names.
* All characters unique in all identifiers.
* Character arrays can be initialized with string literals.
* Handles the multi-dimensional array bug in MW C.
* Recognizes standard preprocessor directives for compatibility with ANSI
  preprocessors.

 * Allows typedefs of incomplete structs and unions.
 * Removes distinction between complete and incomplete structs and unions.

Known bugs:
 * Doesn't detect if a goto uses a label that is never defined.
 * Allows new struct/union specifictions in function signatures.
 * Doesn't handle const typedefs properly.
 * Doesn't handle casts of constants properly.

To do:
 * fix the bugs
 * implement bit fields
 * include source text in error messages
 * implement unsigned long
 * Add warning messages for questionable constructs
 * Add warning messages for non-portable constructs

Changes since v0.1
 * Line numbers on error messages are now displayed relative to the original
   source file.
 * goto and label statements are now recognized.
 * fixed a bug where a pointer subscripted with a constant value was
   considered constant.
 * preprocessor directives are no longer stripped.
 * "direct" declarations are now allowed.
 * Variable redeclaration is now checked.
 * Declarations are now checked against forward declarations.
 * Output lines are now kept from being more than 80 chars long.
 * Added support for elipsis in prototype form function signatures.
 * Added support for structure passing with old style function parameters.

Changes since v0.2
 * Fixed a problem in defining functions with elipsis in their signature.
 * Added support for enumerations.
 * Array parameters are now converted to pointers properly.
 * Fixed a problem with function calls with constants as their last
   argument.

Changes since v0.3
 * Field names are now encoded so conflicts are impossible.
 * All identifiers not externally visible that are more than eight characters

```
   long are encoded to eight characters to prevent ambiguity.
 *  xternally visible identifier names more than eight characters long are
   hashed to eight characters to make ambiguity very unlikely.
 * Character arrays can now be initialized with string literals.
   The string is converted to a sequence of individual characters.
 * Aggregate initializers are now checked for correctness.
 * Arrays are cast to their appropriate type when dereferenced to account
   for a bug in MW C.
 * Hex constant evaluation fixed.
 * Array dimensions can now be arbitrary constant expressions.
 * Code generated for auto arrays with aggregate initializers when the
   primary array dimension isn't specified is now correct.

Changes since v0.
 * Code generated for for loops fixed.  If a continue was in the loop,
   the third expression of the for statement would not be executed.

Changes since v0.5
 *  ixed bug with generating incorrect values for certain escape sequences.
 * Added recognition of #line, #pragma edn, #pragma asm, #pragma src.
 *  ixed bug with prototypes using typedef names.
 *  ixed many other bugs and improved error reporting in several cases.

Changes since v0.
 * typedefs of incomplete struct/unions implemented.
 *  istinction between structs before and after definition removed.
 *  ixed several problems with floating point numbers.
 *  ixed bug with ' ' operator.
 * Added checks for undefined variables in several places.
 *  ixed a problem with the "do-while" construct.
 * Made "struct direct" allowable.

Changes since v0.
 * Support for 'L' suffix on integers.
 * Support for constant expressions involving longs.
 *  ixed bug with function pointers in structs.
 * Made calling a function pointer without indirection possible.
 * Many small bug fixes.

Changes since v0.
 * Many small bug fixes.
```

 * ore efficient use of memory.

Changes since v0.
 * Fixed a bug with "direct" arrays.
 * Added the output of .ntbl files.

Changes since v0.10
 * Fixed a bug with arrays of struct pointers.

Changes since v0.11
 * roblem with initialized auto arrays fixed.
 * ow checks for functions declared extern then redeclared static.
 * Assignment of a number to an array is now caught as an error.
 * Assignment of structures of different types is now caught as an error.
 * void is now never allowed to be assigned
 * andling of arithmetic operations on non-int constants is handled better.
 * A function with a void parameter list used to be able to be called with
   more than 0 parameters, but this is now disallowed.
 * ome syntax errors are now fatal (no error recovery attempted) since
   error recovery was rarely successful.
 * ntbl files are now output using the original source file name.   efore,
   ntbl files would be created for header files, but this was annoying
   since the linker stated the original source file name in error messages.

 lease let me know if you have any problems or suggestions.   y email
address is gcato st 000.sct.edu.

  ntil Later,
 Vaughn Cato