NAME

occ2c – Translator from Occam Source into C

SYNOPSIS

occ2c [options] file

Options: [–dghHilsuvw] [–o <file>] [–m[acefilmnprvw]] [–e[s|t|c] [[–t<number>] [–p[acopts]] [–z[cdelnqstvx]]

DESCRIPTION

This manual page documents versions 0.9b and later of **occ2c,** the Occam to C translator of the **SPOC (Southampton Portable Occam Compilation) system.** The standard action of the command

occ2c <file>.occ

is to take the occam program contained in *<file>.occ* and produce an ANSI C version of the program in *<file>.c* The default settings of all the various options can be seen with the **-h** option.

The translation system internally consists of several phases:
Lexical Analysis and Parsing
Type Checking
Attribute evaluation
Code Transformation
Code Generation.

TRANSLATION AND COMPILATION

An Occam program consists of a number of code modules and libraries. The outer-level code module (main module) may reference any number of library units, which can be sub-modules of the same program, or actual libraries.

ENVIRONMENT

occ2c requires a number of predefined header and template files to be available, both for internal usage, and for include and library files. The defaults for these files are located through the *SPOC* environment variable which should be directed towards the root directory of the compiler installation. These files can be superceded either by placing them in the current working directory or by adding an *OSEARCH* environment variable containing a sequence of space-seperated paths, each terminated by a slash. The search order is current working directory, left to right through the directies listed in *OSEARCH* and finally default locations relative to the *SPOC* environment variable.

GENERAL OPTIONS

–d Enable inline C directives in the Occam source. These are generally used in library files for access to host i/o facilities. There are two directives,

#H <line_of_C_code>

which is only allowed at the outermost level (typically for include C header files) and

#C <line_of_C_code>

which is valid anywhere an occam process is allowed. Simple occam variables can be used within the C code by prefixing them with a

−o <file>
Outputs the generated C code to the *<file>*, the filename – can be used direct the code to standard out.

−s This option forks off an SP server as a separate Unix process which talks SP protocol via a socket to the SP channels passed in to the top of an application. This gives full inmos iserver compatibility but degrades IO performance and introduces a dependence on the proprietary Inmos libraries. When used versions of the spserver.lib and the associated proprietary Inmos libraries MUST APPEAR on the OSEARCH path.

−t<number>
Sets tab to be equivalent to *<number>* spaces, the default is 8. If no number is supplied then the option toggles whether tabs are allowed in the occam source at all.

−u Generates warnings for usage violations, rather than errors. This allows translation to continue to code generation, despite the presence of a range of errors including parallel usage errors (such as writing to a variable in multiple parallel threads) and alias errors (such as writing to an abbreviated variable/array).

−v Generate verbose translation information, including the translator version and build date, and messages describing the various translation phases as they are performed.

−w Suppress warning messages. No warning messages will be displayed, only errors.

CODE GENERATION OPTIONS

−e[s|t|c]
Select error mode. Occam programs handle errors in one of three ways, either ignoring the error, stopping the erroneous process, or terminating the application. The error mode is selected by a suffix letter:
s – Stop process on error
t – Terminate application on error (the default)
c – Continue on error (ignore errors)

−g Enable source level Occam debugging.

−l Enable/Disable generation of library units (as opposed to a main module).

−m Change code generation mode, suffixed by any combination of:
a - disable Array checks
c - disable Conversion checks
f - enable calls to NAG F90
m - enable coMmunication checks
r - disable Range checks
n - disable generatation of occam line Numbers
e - does all RETYPE constant folding assuming a target of OPPOSITE endianness to the machine running the compiler.
i - generate target independent code to perform RETYPES at run-time
l - enable INT64 support by using GNU C's *long long* data type.
w - use default INT size of 16-bit. BEWARE code generated this way CANNOT be used with any modules/libraries compiled for 32-bits.
p - enable PLACED PAR support
v - enable free variable access from within PLACED PAR

INTERNAL TRANSLATOR DEBUGGING OPTIONS

−i Enable / Disable automatic inclusion of standard Occam intrinsic function library. This option is enabled by default, but must be disabled to compile the intrinsic library itself.

−p Control internal compilation phases. Suffix by any combination of:
- a - disable c output Attribute calculations
- c - disable c code generation
- o - enable occam code generation (to standard out)
- p - disable parallel usage Attribute calculations
- t - disable typecheck Attribute calculations
- s - disable code Simplification/transformation

−z Enable/Disable internal debugging options, suffix by any combination of:
- c - Enable CodeGenerator Debugging
- d - Debug parser
- e - Sort error messages (enabled by default)
- l - Lex only
- n - output Occam line numbers as comments within generated C
- q - Query syntax tree
- s - show Symbol table
- t - show abstract syntax Tree
- v - show variable declaration details
- x - leXical debug