Gontc manpage

August 28, 2002

1 NAME

gontc - compile and link Gont files

2 SYNOPSIS

gontc [*OPTIONS*] *files...*

3 DESCRIPTION

gontc is used to compile source Gont intro binaries, and then linking them together. It uses concept of module in managing compilation. Module consists of interface (in *.gi source file), and implementation (in *.g source file). In order to compile implementation compiler has to be able to find compiled interface for module being compiled, and all modules it depends on.

Typical compilation involves running, for each module in project, **gontc -c module.gi** to compile interfaces, then **gontc -c module.g** to compile implementation, and finally **gontc -o prog mod1.xgo mod2.xgo...**

4 OPTIONS

gontc recognizes following options:

- **-c**
  just compile, don't try to link
- **-ksi** *OPT*
  pass option to Ksi compiler. Example usage might be: **-ksi -fomit-frame-pointer**
- **-ksicc** *PROG*
  set Ksi compiler, for example **-ksicc ppc-pld-linux-cc**
- **-O***
  Same as -ksi -O* (where * stands for any string).
- **-g***
  produce debug info. Same as -ksi -g*.
- **-f***
  Same as -ksi -f*.

- **-m***

  Same as -ksi -m*. Example: -march=athlon.

- **-pg**

  Same as -ksi -pg.

- **-S**

  just generate *.ksi file and stop.

- **-o** *OUTFILE*

  set linker output file name. This doesn't work with **-c**

- **-prefix** *PREFIX*

  override prefix. Prefix is directory in which **gontc** looks for standard libraries.

- **-v**

  be verbose, display what is being done.

- **-a**

  operate in make compiled implementation archive mode. Usage is then: **gontc -a -o** *archive.ga mod1.xgo...*

- **-L** *DIR*

  add directory to library search path. This affects only looking for *.xgo and *.xga files. Use **-ksi -L** is you want to change path for *.a files.

- **-I** *DIR*

  add directory to interface search path. **gontc** looks for compiled interfaces (*.xgi) there.

- **-ciface**

  this option causes **gontc** to output C header files for named module, i.e. subsequent non-option argument is treated as name of modules to dump. Example usage might be: gontc -ciface Foobar ¿ foobar.h.

- **-quote** *ARG*

  treat ARG as ordinary file even if the name starts with "-".

- **-ld** *OPT*

  pass OPT to ksi linker. If used with -a options are stored in library and added when it's linked into executable.

- **-Ftime-report**

  dump timing and memory statistics after compilation.

- **-help**

  display list of options.

- –**help**

  same as **-help**

5 WARNING OPTIONS

gontc also recognizes several options for dealing with turning on/off various warnings. Each of these options can be preceded with "no-" to reverse its meaning. For example -Wno-labeled can be used to turn off warning about unlabeled breaks.

Warning options defaults to "-Wall -Wverbose -Wno-ignored-retval"

- **-Werror**

  treat warnings as fatal errors.

- **-Wnone**

  same as -Wno-all.

- **-Wall**

  enable all warnings:

- **-Wverbose**

  make error messages more verbose.

- **-Wdead**

  warn about unreachable code.

- **-Wlabeled**

  warn about break/continue without label in labeled loop.

- **-Wpattern**

  warn about not exhaustive/redundant pattern matching.

- **-Wcapitalization**

  warn about capitalized pattern variables. If you wonder what might be wrong with pattern variables starting with uppercase letter, consider what will happen if you misspell name of type constructor of even worse exception in pattern matching. It will be treated just as "x" – catch-all pattern.

- **-Wignored-value**

  warn about ignored value of expression. For example about "2+2;". This does not warn about ignored value of '=', '++' and '–' operators. It also doesn't complain about ignored value of function call, see -Wignored-retval for that.

- **-Wignored-retval**

  warn about ignored value of call. This is off by default since it causes problems with abort-like functions.

6 FILES

gontc recognizes type of input file based on extension. Following rules apply:

- ***.g** *Gont sources*

  are compiled into *.ksi and *.xgo. *.ksi file is passed to Ksi compiler, that produces *.o object file with machine code.

- ***.gi** *Gont interface sources*

  are compiled into *.xgi.

- ***.xgo** *Gont compiled implementation file*

  are used to resolve linking dependencies. You should never supply *.o file resulting from *.g compilation for linking. Always use *.xgo file. Otherwise linking error will occur (unresolved reference to Module$$xgo_linked$ or something similar).

- ***.xgi** *Gont compiled interface files*

  gontc doesn't know what to do with this kind of files. I.e. they cannot be passed as command line options. They are open implicitly during compilation. **-I** option can be used to help **gontc** find this kind of files.

- ***.ksi** *Ksi source files*

  are passed to Ksi compiler. It should produce *.o files from them.

- ***.o** *Object files with machine code*

  are just passed to linker. Never use *.o files, produced from *.g, directly. Always use *.go files.

- ***.a** *Libraries of object files with machine code*

  are just passed to linker. Never use *.a files, produced from linking results of *.g compilation together, directly. Always use *.ga files.

- ***.xga** *Archives of Gont implementation files*

  are unpacked to memory, and then treated as if each of *.xgo files inside was specified in command line. If such a file is specified with `:force` appended to it (like `mylib.xga:force`) then all files from this library are linked, and their init/fini sections are executed. It is called "greedy linking" and is useful with helper libraries, when you want entire library to be linked, not only needed files.

7 AUTHOR

This manpage was written by Michal Moskal ¡malekith/at/pld-linux.org¿.