

Линковщик

Используйте команду `nios2-elf-g++` для компоновки ваших объектных файлов и архивов в окончательный исполняемый формат ELF.

Пример использования линковщика

Для компоновки ваших объектных файлов и архивов в **.elf** файл, откройте командную среду Nios II и вызовите `call nios2-elf-g++` с соответствующими аргументами. В следующем примере показан вызов линковщика:

```
nios2-elf-g++ -T'<linker script>' -msys-crt0='<crt0.o file>' \  
-msys-lib=<system library> -L '<The path where your libraries reside>' \  
-DALT_DEBUG -O0 -g -Wall -mhw-mul -mhw-mulx -mno-hw-div \  
-o <your project>.elf <object files> -lm ↵
```

Поведение линковщика для вашего проекта может отличаться. Когда вы собираете проект в Nios II IDE, вы можете видеть командную строку, используемую для компоновки вашего приложения. Чтобы включить эту опцию в Nios II IDE, в меню Window кликните **Preferences** и выберите вкладку **Nios II**, где разрешите **Show command lines when running make**. Вы можете также напрямую указать отображение в командной строке при запуске `make` (делать) без опции `-s` из командной среды Nios II.

Altera рекомендует вам не использовать встроенный линковщик `nios2-elf-ld` для компоновки вашей программы. Для процессора Nios II, как и для всех программных процессоров, процесс компоновки является комплексным. Командная опция `g++` (`nios2-elf-g++`) упрощает этот процесс. Большинство опций задаётся опцией командной строки `-m`, однако доступные опции зависят от выбора процессора.

`nios2-elf-size`

Эта команда отображает общий размер вашей программы и её основных секций кода.

Пример использования `nios2-elf-size`

Для отображения информации о размере вашей программы выполните следующие пункты:

1. Откройте командную среду Nios II.
2. Смените директорию на директорию размещения вашего объектного файла.
3. В командной строке введите следующую команду:

```
nios2-elf-size <project>.elf
```

В примере 4-9 показана информация о размере, предоставляемая этой командой.

Example 4-9. `nios2-elf-size` Command Usage

```
[SOPC Builder]$ nios2-elf-size my_project.elf
text      data      bss      dec      hex filename
272904     8224 6183420 6464548 62a424 my_project.elf
```

`nios2-elf-strings`

Эта команда отображает все строки в `.elf` файле.

Пример использования `nios2-elf-strings`

Команда имеет единственный аргумент:

```
nios2-elf-strings <project>.elf
```

`nios2-elf-strip`

Эта команда разбирает все символы из объектных файлов. Поддерживаются все объектные файлы, включая ELF, объектные файлы (`.o`) и архивные файлы (`.a`).

Пример использования `nios2-elf-strip`

```
nios2-elf-strip <options> <project>.elf
```

Примечания к использованию `nios2-elf-strip`

Команда `nios2-elf-strip` уменьшает размер `.elf` файла. Эта команда полезна только когда процессор Nios II запускается на операционной системе со встроенной поддержкой ELF. Если ELF является встроенным исполняемым форматом, соответствующий `.elf` файл храниться в памяти, и его размер имеет значение. Если нет, то файл анализируется, а его инструкции и данные хранятся прямо в памяти без символов. Linux является одной из операционных систем со встроенной поддержкой ELF; uClinux - другой. uClinux использует исполняемый формат (FLT), который транслируется прямо из ELF.

nios2-elf-gdbtui

Эта команда запускает сессию GDB в которой терминал отображает исходный код рядом с обычной GDB консолью.

Синтаксис для команды nios2-elf-gdbtui идентичен синтаксису команды nios2-elf-gdb, описанной в секции "nios2-elf-gdb" на стр. 4-15.

При использовании отладчика Nios II GDB доступны два дополнительных пользовательских интерфейса GDB. CGDB - курсорный GDB UI - доступен с <http://sourceforge.net/>. Экран данных отладчика (DDD) - очень рекомендован.

nios2-elf-gprof

Эта команда позволяет вам профилировать вашу Nios II систему.

За подробной информацией об этой команде и об окончательном виде графической оболочки Nios II IDE, обратитесь к [AN 391](#): *Профилирование систем Nios II*.

nios2-elf-insight

Команда nios2-debug, описанная в секции "nios2-debug" на стр. 4-12 использует эту команду для начала сессии отладчика Insight для соответствующего .elf файла.

nios2-elf-gcc и g++

Эти команды запускают компилятор GNU C или C++ для процессора Nios II.

Пример использования команд компиляции

В следующем простом примере показана командная строка, запускающая компилятор GNU C или C++:

```
nios2-elf-gcc(g++) <options> -o <object files> <C files>
```

Более сложный пример компиляции

В примере 4-10 показана командная строка, сгенерированная в Nios II IDE, которая компилирует Си код в несколько файлов в нескольких директориях.

Example 4-10. Example nios2-elf-gcc Command Line

```

nios2-elf-gcc -xc -MD -c \
-DSYSTEM_BUS_WIDTH=32 -DALT_NO_C_PLUS_PLUS -DALT_NO_INSTRUCTION_EMULATION \
-DALT_USE_SMALL_DRIVERS -DALT_USE_DIRECT_DRIVERS -DALT_PROVIDE_GMON \
-I. -I/cygdrive/c/Work/Projects/demo_reg32/Designs/std_2s60_ES/software/\
reg_32_example_0_syslib/Release/system_description \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_timer/HAL/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_timer/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_jtag_uart/HAL/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_jtag_uart/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_pio/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_lcd_16207/HAL/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_lcd_16207/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_sysid/HAL/inc \
-I/cygdrive/c/altera/70_b31/ip/sopc_builder_ip/altera_avalon_sysid/inc \
-I/cygdrive/c/altera/70_b31/nios2eds/components/altera_nios2/HAL/inc \
-I/cygdrive/c/altera/70_b31/nios2eds/components/altera_hal/HAL/inc \
-DALT_SINGLE_THREADED -D_hal_ -pipe -DALT_RELEASE -O2 -g -Wall \
-mhw-mul -mhw-mulx -mno-hw-div -o obj/reg_32_buttons.o ../reg_32_buttons.c

```

nios2-elf-c++filt

Эта команда восстанавливает испорченные имена C++. Синтаксис C++ позволяет нескольким функциям иметь одинаковые имена, тогда как их параметры могут отличаться; чтобы отслеживать каждую конкретную функцию, компилятор портит или изменяет имена функций. Каждый компилятор портит функции на свой манер.

За подробной информацией, включая подробную информацию о том, как различные компиляторы портят имена функций C++, обратитесь к стандартной справке для C++ компиляторов.

Пример использования nios2-elf-c++filt

Чтобы посмотреть оригинальное, восстановленное имя функции, которая относится к соответствующему имени символа, вы можете ввести следующую команду:

```

nios2-elf-c++filt -n <symbol name> ↩
Например:
nios2-elf-c++filt -n _Z11my_functionv ↩

```

Более сложный пример nios2-elf-c++filt

В следующем примере из командной строки вызывается отображение всех восстановленных имён функций из выбранного файла:

```

nios2-elf-strings <file>.elf | grep ^_Z | nios2-elf-c++filt -n

```

В этом примере операция nios2-elf-strings выводит все строки из .elf файла. Результат отправляется в операцию grep, которая идентифицирует все строки, начинающиеся с _Z. (GCC всегда присоединяет к испорченным именам функций _Z). Результат команды grep отправляется в команду nios2-elf-c++filt. Результатом которой является список всех восстановленных функций в GCC C++ .elf файле.

nios2-elf-nm

Эта команда выводит все символы в **.elf** файле.

Пример использования *nios2-elf-nm*

Есть два простых примера, показывающих использование команды *nios2-elf-nm*:

- `nios2-elf-nm <project>.elf` ➡
- `nios2-elf-nm <project>.elf | sort -n` ➡

Более сложный пример *nios2-elf-nm*

Для генерирования списка символов из вашего **.elf** файла в порядке возрастания, используйте следующую команду:

```
nios2-elf-nm <project>.elf | sort -n > <project>.elf.nm
```

Файл **<project>.elf.nm** содержит все символы из вашего исполняемого файла, представленные в порядке возрастания. В этом примере, команда *nios2-elf-nm* создаёт список символов. В тексте этого списка адрес каждого символа является первым полем в новой строке. Опция **-n** для команды сортировки задаёт сортировку по адресу в цифровом виде, взамен сортировки по алфавиту (по умолчанию).

nios2-elf-objcopy

Используйте эту команду для конвертирования из одного бинарного объектного формата в другой, дополнительно изменяя бинарные данные.

Команда *objcopy* не ограничивается конвертацией из и в ELF файлы. Вы можете использовать эту команду для конвертации в любые форматы, представленные в табл. 4-3.

Табл. 4-3. Бинарные форматы *-objcopy*

Команда (...-objcopy)	Комментарии
elf32-littlenios2, elf32-little	Заголовок little endian, данные little endian. Часто используемый формат по умолчанию.
elf32-bignios2, elf32-big	Заголовок big endian, данные big endian.
srec	S-Record (SREC) формат выхода
symbolsrec	Формат SREC со всеми символами, представленными в заголовочном файле, предшествующими SREC данным.
tekhex	Шестнадцатеричный формат Tektronix (TekHex)
binary	Необработанный бинарный формат. Прекрасно подходит для создания бинарных образов для сохранения во флеш памяти в вашей встроенной системе.
ihex	Шестнадцатеричный формат Intel (ihex)

Вы можете получить информацию о файлах бинарных форматов (TekHex, ihex и прочих) в интернете. Вначале вы можете узнать об этих форматах на сайте www.sbprojects.com, введя необходимый формат файла.

Пример использования *nios2-elf-objcopy*

Для создания SREC файла из ELF файла, используйте следующую команду:

```
nios2-elf-objcopy -O srec <project>.elf <project>.srec
```

ELF является предполагаемым бинарным форматом, если другое не представлено. За информацией о том, как задать другой бинарный формат в командной среде Nios II введите следующую команду:

```
nios2-elf-objcopy --help
```

nios2-elf-objdump

Используйте эту команду для отображения информации об объектном файле, обычно об ELF файле. Команда *nios2-elf-objdump* поддерживает все бинарные форматы, поддерживаемые командой *nios2-elf-objcopy*, но ELF - это единственный формат, поддерживающий все опции командной строки.

Пример использования *nios2-elf-objdump*

Nios II IDE использует следующую команду для генерирования дампа объектного файла:

```
nios2-elf-objdump -D -S -x <project>.elf > <project>.elf.objdump
```

nios2-elf-ranlib

Вызов функции *nios2-elf-ranlib* эквивалентен вызову *nios2-elf-ar* с опцией *-s* (*nios2-elf-ar -s*). За дополнительной информацией об этой команде, обратитесь к секции "nios2-elf-ar" на стр. 4-16 или введите *nios2-elf-ar --help* в командной среде Nios II.