

7. Средства автоматической сборки проектов

Разделы:

- Особенности использования GNU *Make*
- Автоматическое конфигурирование
- Автоматическое создание *make*-файла



Введение

- Утилита *make* автоматизирует процесс компиляции программ любого размера и сложности так, что одна единственная команда *make* заменяет сотни команд компиляции и компоновки
- *make* сравнивает временные штампы связанных файлов во избежание повторения уже выполненной работы
- *make* управляет индивидуальными правилами, которые определяют, как строить различный целевой код и автоматически анализирует зависимости между используемыми файлами

Цели, зависимости и команды

```
# A basic makefile for "circle".
```

```
CC = gcc
```

```
CFLAGS = -Wall -g -std=c99
```

```
LDFLAGS = -lm
```

```
circle : circle.o circulararea.o
```

```
$(CC) $(LDFLAGS) -o $@ $^
```

```
circle.o : circle.c
```

```
$(CC) $(CFLAGS) -o $@ -c $<
```

```
circulararea.o: circulararea.c
```

```
$(CC) $(CFLAGS) -o $@ -c $<
```

Цели, зависимости и команды

- Общий синтаксис:

```
цель1 [цель2 ...] :[:] [зависимости1 ...]  
      [; команды] [# ...] [\t команды ] [# ...]
```

```
# etags не запустится из src
```

```
TAGS:
```

```
    cd src/
```

```
    etags *.c
```

```
# etags запустится из src
```

```
TAGS:
```

```
cd src/ ; \
```

```
etags *.c
```

Цели, зависимости и команды

```
# A basic makefile for "circle".
```

```
CC = gcc
```

```
CFLAGS = -Wall -g -std=c99
```

```
LDFLAGS = -lm
```

```
circle : circle.o circulararea.o
```

```
$(CC) $(LDFLAGS) -o $@ $^
```

```
circulararea.o circle.o: %.o: %.c
```

```
$(CC) $(CFLAGS) -o $@ -c $<
```

Цели, зависимости и команды

- Общий вид шаблонных правил:

```
[target_list :] target_pattern : prerequisite_pattern  
[command-script]
```

```
%.o: %.c
```

```
$(CC) $(CFLAGS) -o $@ -c $<
```

- имя файла *app%3atmodule.o* соответствует шаблону *app\%3a%.o*
- имя файла *app\module.o* должно соответствовать шаблону *app\\%.o*

Цели, зависимости и команды

```
# A slightly more elegant makefile for "circle".
```

```
CC = gcc
```

```
CFLAGS = -Werror -std=c99
```

```
OBJS = circle.o circulararea.o
```

```
circle: $(OBJS) -lm
```

Цели, зависимости и команды

```
...  
# default  
OUTPUT_OPTION = -o $@  
# default  
LINK.o = $(CC) $(LDFLAGS) $(TARGET_ARCH)  
# default  
COMPILE.c = $(CC) $(CFLAGS) $(CPPFLAGS) $(TARGET_ARCH) -c  
%: %.o  
# commands to execute (built-in):  
$(LINK.o) $^ $(LOADLIBES) $(LDLIBS) -o $@  
%.o: %.c  
# commands to execute (built-in):  
$(COMPILE.c) $(OUTPUT_OPTION) $<  
...
```


Цели, зависимости и команды

```
%: %.o
```

```
cc -o $@ $^
```

```
%.o: %.c
```

```
cc -c -o $@ $<
```

```
$ ls
```

```
Makefile square.c
```

```
$ make -r square
```

```
cc -c -o square.o square.c
```

```
cc -o square square.o
```

```
rm square.o
```

```
$ ls
```

```
Makefile square square.c
```

Правила с двойным двоеточием

```
target :: prerequisites
      commands
```

```
# A makefile for "circle" to demonstrate double-colon rules.
```

```
CC = gcc
RM = rm -f
CFLAGS = -Wall -std=c99
DBGFLAGS = -ggdb -pg
DEBUGFILE = ./debug
SRC = circle.c circulararea.c
```

```
circle :: $(SRC)
    $(CC) $(CFLAGS) -o $@ -lm $^
```

```
circle :: $(DEBUGFILE)
    $(CC) $(CFLAGS) $(DBGFLAGS) -o $@ -lm $(SRC)
```

```
.PHONY : clean
clean :
    $(RM) circle
```

Переменные

- Оператор присваивания '=' мы используем для рекурсивно-подставляемой переменной

```
DEBUGFLAGS = $(CFLAGS) -ggdb -DDEBUG -O0
```

- Для создания просто подставляемой переменной мы используем оператор ':='

```
OBJ = circle.o circulararea.o
```

```
TESTOBJ := $(OBJ) profile.o
```

- В командной строке:

```
$make CFLAGS=-ffinite-math-only  
circulararea.o
```

Переменные

- Любая переменная может быть отменена оператором присваивания внутри *make*-файла, если оператор начинается с ключевого слова *override*

```
override CPPLFLAGS = -DDEBUG
```

- Использовать такую форму оператора присваивания следует с осторожностью
- Есть еще две разновидности присваивания: ``+='` и ``?='`
- Первая – это оператор добавления, вторая – оператор условного присваивания
- См. в соответствующих справочных источниках

Переменные

```
ONESPACE := ' '
```

```
TEST = Does$(ONESPACE) this$(ONESPACE) work?
```

```
NOTHING :=
```

```
ONESPACE := $(NOTHING) # This comment  
terminates the variable's value.
```

- Мы можем выполнить любую из операций присваивания применительно к определенной цели включением в *make*-файл строки следующей формы:

```
target_list: [override] assignment
```

Переменные

```
# Tools and options:
CC = gcc
CFLAGS = -c -Wall -std=c99 $(ASMFLAGS)
DEBUGCFLAGS = -ggdb -O0
RM = rm -f
MKDIR = mkdir -p
# Filenames:
OBJ = circle.o circulararea.o
SYMTABS = $(OBJ:.o=.sym)
EXEC = circle
# The primary targets:
production: clean circle
testing: clean debug
symbols: $(SYMTABS)
clean:
    $(RM) $(OBJ) *.sym circle circle-dbg
# Rules to build prerequisites:
circle debug: $(OBJ) -lm
    $(CC) $(LDFLAGS) -o $(EXEC) $^
$(OBJ): %.o: %.c
    $(CC) $(CFLAGS) $(CPPFLAGS) -o $@ $<
$(SYMTABS): %.sym: %.c
    $(CC) $(CFLAGS) $(CPPFLAGS) -o $*.o $<
# Target-specific options:
debug: CPPFLAGS += -DDEBUG
debug: EXEC = circle-dbg
debug symbols: CFLAGS += $(DEBUGCFLAGS)
symbols: ASMFLAGS = -Wa,-as=$*.sym,-L
```

Переменные

```
$ (name:ending=new_ending)
```

- Когда мы таким способом ссылаемся на переменную, *make* делает подстановку, потом проверяет конец каждого слова в проверяемом значении до завершения строки
 - Если слово заканчивается *ending*-ом, то *make* заменяет эту часть *new_ending*-ом
-

```
gcc -c -Wall -std=c99 -Wa,-as=circle.sym,-L -ggdb -O0 -o circle.o circle.c  
gcc -c -Wall -std=c99 -Wa,-as=circulararea.sym,-L -ggdb -O0 -o  
circulararea.o circulararea.c
```

Переменные

- \$@ - Имя файла цели
- \$* - Основа имени файла цели, т.е. часть, представленная метасимволом % в шаблонном правиле
- \$< - Первая зависимость
- \$^ - Список зависимостей, исключая дублирующиеся элементы
- \$? - Список зависимостей, которые новее цели
- \$+ - Полный список зависимостей, включая дубликаты
- \$% - Если цель является элементом архива, то переменная \$% позволяет получить имя элемента без указания имени архива, а для получения последнего предназначена \$@

Переменные

- У *make* есть специальная нотация для целей, которые являются элементами архива:

```
archive_name(member_name) : [prerequisites]  
[command_script]
```

- Имя элемента архива заключается в круглые скобки немедленно после имени файла-архива

```
AR = ar -rv
```

```
libcircularmath.a(circulararea.o) :  
    circulararea.o
```

```
$(AR) $@ $%
```

- Это правило выполняется, когда следующая команда добавляет или удаляет объектный файл из архива:

```
ar -rv libcircularmath.a circulararea.o
```

Переменные

- Когда автоматические переменные преобразуются в список, элементы отделяются пробелами
- Чтобы отделить имена файлов от каталогов, существуют еще две версии каждой автоматической переменной в списке, чьи имена формируются суффиксами *D* и *F*
- Поскольку результирующее имя переменной состоит из двух символов, требуются круглые скобки
- Например, $\$(@D)$ в любом правиле заменяется на имя каталога для заданной цели, тогда как $\$(@F)$ дает только имя файла без имени каталога

Переменные

- *MAKEFILES* – стандартные имена *make*-файлов
- *MAKEFILE_LIST* – список всех файлов, задействованных в текущем активном сценарии
- *MAKELEVEL* – Уровень вложенности для запускаемых рекурсивно *make*-файлов
- *MAKEFLAGS* – Список аргументов, с которыми запускался *make*-файл
- *MAKECMDGOALS* – Список целей, с которыми запускался *make*-файл
- *CURDIR* – Имя текущего рабочего каталога
- *VPATH* – Имя каталога, в котором *make* ищет файлы, если их нет в *CURDIR*
- *SHELL* – имя языка командной оболочки
- *SUFFIXES* – список известных суффиксов для суффиксных правил
- и др.

ФИКТИВНЫЕ ЦЕЛИ

```
.PHONY: clean
```

```
.PHONY: bin
```

```
bin: circle
```

```
$(MKDIR) $@
```

```
$(CP) $< $@/
```

```
$(CHMOD) 600 $@/$<
```

```
bin: circle
```

```
$(MKDIR) $@
```

```
$(CP) $< $@/
```

```
$(CHMOD) 600 $@/$<
```

```
bin:
```

ФИКТИВНЫЕ ЦЕЛИ

- **.PRECIOUS**

- В нормальном случае когда выполнение *make*-файла прервано, то цель *make* удаляет цель
- Если указать данный атрибут, то объявленная цель не удаляется ни при каких обстоятельствах

- **.INTERMEDIATE**

- Когда требуется построить некую цель, то сначала строятся ее зависимости
- Для явно отсутствующих целей *make* ищет неявные правила для их построения
- Они называются **промежуточными**
- По умолчанию удаляются, поведение меняется указанием данного атрибута

- **.SECONDARY**

- Похож на **.INTERMEDIATE**, но промежуточные цели не удаляются

- **.IGNORE**

- Игнорирование возникающих ошибок

- **.LOW_RESOLUTION_TIME**

- Изменение длительности устаревания файлов



GNU *Autotools*

- Для сборки из исходных текстов обычно выполняются

\$ *./configure*

\$ *make*

\$ *make install*

- К *Autotools* относятся
 - *Autoconf*
 - *Automake*
 - *Libtool*
 - *Shtool*

Autoconf

```
/* hello.c: A standard "Hello!" program */

#include <stdio.h>

int main(int argc, char* argv[])
{
    printf("Hello, goodbye and farewell!\n");

    return 0;
}
-----
# Makefile: A standard Makefile for hello.c

all: hello

clean:
    rm -f hello *.o
-----
$ ls
Makefile  hello.c
$ make
gcc      hello.c  -o hello
$ ls
Makefile  hello  hello.c
$ ./hello
Hello, goodbye and farewell!
$
```

Autoconf

```
$ autoscan
$ ls
Makefile  autoscan.log  configure.scan  hello  hello.c
$
-----
$ mv configure.scan configure.ac
-----
$ autoconf
$ ls
Makefile      autoscan.log  configure.ac  hello.c
autom4te.cache/ configure      hello
$
-----
$ mv Makefile Makefile.in
$ ls
Makefile.in      autoscan.log  configure.ac  hello.c
autom4te.cache/  configure      hello
$
```


Autoconf

```
$ ./configure
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: error: cannot find input file: config.h.in
$ make clean all
rm -f hello
gcc      hello.c      -o hello
$ ls
Makefile      autom4te.cache/  config.log      configure      hello
Makefile.in   autoscan.log     config.status   configure.ac   hello.c
$ ./hello
Hello, goodbye and farewell!
$
```

Autoconf

```
$ ls
Makefile      autom4te.cache/  config.log      configure      hello
Makefile.in   autoscan.log     config.status   configure.ac   hello.c
$ autoheader
autoheader2.50: `config.h.in' is created
$ ls
Makefile      autoscan.log     config.status   hello
Makefile.in   config.h.in      configure       hello.c
autom4te.cache/ config.log       configure.ac
$ ./configure
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
$ ls
Makefile      autoscan.log     config.log      configure.ac
Makefile.in   config.h         config.status   hello
autom4te.cache/ config.h.in      configure       hello.c
$
```

Autoconf

```
/* config.h.  Generated by configure.  */
/* config.h.in.  Generated from configure.ac by autoheader.  */

/* Define to the address where bug reports for this package should be sent. */
#define PACKAGE_BUGREPORT "BUG-REPORT-ADDRESS"

/* Define to the full name of this package. */
#define PACKAGE_NAME "FULL-PACKAGE-NAME"

/* Define to the full name and version of this package. */
#define PACKAGE_STRING "FULL-PACKAGE-NAME VERSION"

/* Define to the one symbol short name of this package. */
#define PACKAGE_TARNAME "full-package-name"

/* Define to the version of this package. */
#define PACKAGE_VERSION "VERSION"
```

Autoconf

```
/* hello.c: A program to show the time since the Epoch */  
  
#include <stdio.h>  
#include <sys/time.h>  
  
int main(int argc, char* argv[])  
{  
    double sec;  
    struct timeval tv;  
  
    gettimeofday(&tv, NULL);  
    sec = tv.tv_sec;  
    sec += tv.tv_usec / 1000000.0;  
  
    printf("%f\n", sec);  
  
    return 0;  
}
```

Autoconf

```
$ autoscan
Use of uninitialized value in concatenation (.) or string at /usr/bin/autoscan line 195.
configure.ac: warning: missing AC_CHECK_FUNCS([gettimeofday]) wanted by: hello.c:12
configure.ac: warning: missing AC_CHECK_HEADERS([sys/time.h]) wanted by: hello.c:4
configure.ac: warning: missing AC_HEADER_TIME wanted by: hello.c:10
$ mv configure.scan configure.ac
$ autoconf
$ autoheader
autoheader2.50: `config.h.in' is updated
$ ./configure
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking sys/time.h usability... yes
checking sys/time.h presence... yes
checking for sys/time.h... yes
checking whether time.h and sys/time.h may both be included... yes
checking for gettimeofday... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
$ ls
Makefile      autoscan.log  config.h.in~  configure     hello.c
Makefile.in   config.h      config.log    configure.ac
autom4te.cache/ config.h.in  config.status  hello
$
```

Autoconf

```
/* config.h.  Generated by configure.  */
/* config.h.in.  Generated from configure.ac by autoheader.  */

/* Define to 1 if you have the `gettimeofday' function. */
#define HAVE_GETTIMEOFDAY 1

/* Define to 1 if you have the <inttypes.h> header file. */
#define HAVE_INTTYPES_H 1

/* Define to 1 if you have the <memory.h> header file. */
#define HAVE_MEMORY_H 1

/* Define to 1 if you have the <stdint.h> header file. */
#define HAVE_STDINT_H 1

/* Define to 1 if you have the <stdlib.h> header file. */
#define HAVE_STDLIB_H 1

/* Define to 1 if you have the <strings.h> header file. */
#define HAVE_STRINGS_H 1

/* Define to 1 if you have the <string.h> header file. */
#define HAVE_STRING_H 1

/* Define to 1 if you have the <sys/stat.h> header file. */
#define HAVE_SYS_STAT_H 1

/* Define to 1 if you have the <sys/time.h> header file. */
#define HAVE_SYS_TIME_H 1

/* Define to 1 if you have the <sys/types.h> header file. */
#define HAVE_SYS_TYPES_H 1
```

Autoconf

```
/* Define to 1 if you have the <unistd.h> header file. */
#define HAVE_UNISTD_H 1

/* Define to the address where bug reports for this package should be sent.
   */
#define PACKAGE_BUGREPORT "BUG-REPORT-ADDRESS"

/* Define to the full name of this package. */
#define PACKAGE_NAME "FULL-PACKAGE-NAME"

/* Define to the full name and version of this package. */
#define PACKAGE_STRING "FULL-PACKAGE-NAME VERSION"

/* Define to the one symbol short name of this package. */
#define PACKAGE_TARNAME "full-package-name"

/* Define to the version of this package. */
#define PACKAGE_VERSION "VERSION"

/* Define to 1 if you have the ANSI C header files. */
#define STDC_HEADERS 1

/* Define to 1 if you can safely include both <sys/time.h> and <time.h>. */
#define TIME_WITH_SYS_TIME 1
```

Autoconf

```
/* hello.c: A program to show the time since the Epoch */
#include <stdio.h>
#include "config.h"

#ifdef HAVE_SYS_TIME_H
#include <sys/time.h>
#else
#include <time.h>
#endif

double get_sec_since_epoch()
{
    double sec;

    #ifdef HAVE_GETTIMEOFDAY
        struct timeval tv;
        gettimeofday(&tv, NULL);
        sec = tv.tv_sec;
        sec += tv.tv_usec / 1000000.0;
    #else
        sec = time(NULL);
    #endif

    return sec;
}

int main(int argc, char* argv[])
{
    printf("%f\n", get_sec_since_epoch());

    return 0;
}
```


Autoconf

```
$ autoscan
$ ./configure
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking sys/time.h usability... yes
checking sys/time.h presence... yes
checking for sys/time.h... yes
checking whether time.h and sys/time.h may both be included... yes
checking for gettimeofday... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: config.h is unchanged
$ make clean all
rm -f hello
cc      hello.c      -o hello
$ ./hello
1053952012.248884
$
```

Automake

```
bin_PROGRAMS=hello
hello_SOURCES=hello.c
-----
$ automake
configure.ac: 4: `automake requires `AM_CONFIG_HEADER', not `AC_CONFIG_HEADER'
automake: configure.ac: `PACKAGE' not defined in `configure.ac'
automake: configure.ac: `VERSION' not defined in `configure.ac'
automake: configure.ac: required file `./install-sh' not found
automake: configure.ac: required file `./mkinstalldirs' not found
automake: configure.ac: required file `./missing' not found
automake: Makefile.am: required file `./INSTALL' not found
automake: Makefile.am: required file `./NEWS' not found
automake: Makefile.am: required file `./README' not found
automake: Makefile.am: required file `./COPYING' not found
automake: Makefile.am: required file `./AUTHORS' not found
automake: Makefile.am: required file `./ChangeLog' not found
configure.ac: 4: required file `./[config.h].in' not found
automake: configure.ac: AC_ARG_PROGRAM must be used in `configure.ac'
automake: configure.ac: AC_PROG_INSTALL must be used in `configure.ac'
$
```

Automake

```
...
#AC_CONFIG_HEADER([config.h])
AM_CONFIG_HEADER([config.h])
...
-----
...
AC_INIT(FULL-PACKAGE-NAME, VERSION, BUG-REPORT-ADDRESS)
AM_INIT_AUTOMAKE(hello, 1.0)
...
-----
$
$ cp -a /usr/share/automake/install-sh .
$ cp -a /usr/share/automake/mkinstalldirs .
$ cp -a /usr/share/automake/missing .
$ touch NEWS README AUTHORS ChangeLog
$ cp -a /usr/share/automake/INSTALL .
$ cp -a /usr/share/automake/COPYING .
$ ls
AUTHORS      Makefile.am      autoscan.log     config.status*   hello.c
COPYING      Makefile.in      config.h          configure*        install-sh*
ChangeLog    NEWS             config.h.in      configure.ac      missing*
INSTALL      README           config.h.in~     configure.scan    mkinstalldirs*
Makefile     autom4te.cache/  config.log        hello*            stamp-h.in
$
-----
AM_CONFIG_HEADER([config.h])
-----
AM_CONFIG_HEADER(config.h)
-----
$ automake
$
```

Automake



```
$ aclocal
```

```
$ ls
```

```
AUTHORS
```

```
COPYING
```

```
ChangeLog
```

```
INSTALL
```

```
Makefile
```

```
Makefile.am
```

```
$
```

```
Makefile.in
```

```
NEWS
```

```
README
```

```
aclocal.m4
```

```
autom4te.cache/
```

```
autoscan.log
```

```
config.h
```

```
config.h.in
```

```
config.h.in~
```

```
config.log
```

```
config.status*
```

```
configure*
```

```
configure.ac
```

```
configure.scan
```

```
hello*
```

```
hello.c
```

```
install-sh*
```

```
missing*
```

```
mkinstalldirs*
```

```
stamp-h.in
```

Automake

```
$ autoconf
$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets ${MAKE}... yes
checking for working aclocal... found
checking for working autoconf... found
checking for working automake... found
checking for working autoheader... found
checking for working makeinfo... found
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking sys/time.h usability... yes
checking sys/time.h presence... yes
checking for sys/time.h... yes
```

Automake

```
checking whether time.h and sys/time.h may both be included... yes
checking for gettimeofday... yes
configure: creating ./config.status
config.status: creating Makefile
config.status: creating config.h
config.status: config.h is unchanged
config.status: executing default-1 commands
$ make
gcc -DHAVE_CONFIG_H -I. -I. -I.      -g -O2 -c hello.c
gcc -g -O2 -o hello hello.o
cd . && autoheader
autoheader2.50: `config.h.in' is updated
cd . \
    && CONFIG_FILES= CONFIG_HEADERS=config.h \
    /bin/sh ./config.status
config.status: creating config.h
config.status: executing default-1 commands
$ ./hello
1053956323.746657
$
```

See also

- *Automake* - <http://www.gnu.org/software/automake/>
- *Make* - <http://www.gnu.org/software/make/>
- *Makefiles. A tutorial by example* - <http://mrbook.org/blog/tutorials/make/>
- *Managing Projects with GNU Make, Third Edition*
By Robert Mecklenburg - <http://oreilly.com/openbook/make3/book/index.csp>
- The GNU configure and build system - <http://airs.com/ian/configure/>
- Работа с утилитой *make* - <http://www.net4me.net/docs/pdf/Linux/make.pdf>
- GNU M4 - <http://www.gnu.org/software/m4/>
- Using *make* and writing Makefiles - <http://www.cs.swarthmore.edu/~newhall/unixhelp/>